

Государственное бюджетное учреждение дополнительного образования  
«Дом юношеского технического творчества»  
Центр цифрового образования «IT-куб» г. Сатка

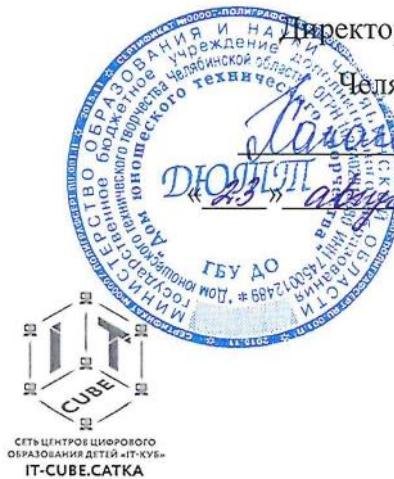
ПРИНЯТО на заседании  
педагогического совета  
ГБУ ДО «ДЮТТ Челябинской  
области»  
протокол № 30 от 22.08 2022

г.

УТВЕРЖДАЮ:

Директор ГБУ ДО «ДЮТТ  
Челябинской области»  
В.Н. Халамов

2022 г.



Дополнительная общеобразовательная общеразвивающая программа  
технической направленности  
**«Программирование на Python»**

Срок освоения программы: 1 год  
Возрастная категория обучающихся: 14 – 18 лет

Автор-составитель:  
Лычковский Константин  
Геннадьевич,  
педагог дополнительного образования

г. Сатка, 2022

## СОДЕРЖАНИЕ

|  |            |
|--|------------|
| ОПИСАНИЕ ПРОГРАММЫ .....   | 3          |
| <b>РАЗДЕЛ 1. КОМПЛЕКС ОСНОВНЫХ ХАРАКТЕРИСТИК ПРОГРАММЫ .....</b>     | <b>5</b>   |
| 1.1 Пояснительная записка .....                                      | 5          |
| 1.2 Цель и задачи программы .....                                    | 7          |
| 1.3 Календарный учебный график .....                                 | 8          |
| 1.4 Учебно-тематический план .....                                   | 9          |
| 1.5 Содержание программы .....                                       | 10         |
| 1.6. Планируемые результаты.....                                     | 13         |
| <b>РАЗДЕЛ 2. КОМПЛЕКС ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИХ УСЛОВИЙ.....</b> | <b>144</b> |
| 2.1 Условия реализации программы .....                               | 144        |
| 2.2 Формы аттестации обучающихся .....                               | 145        |
| 2.3 Оценочные и методические материалы.....                          | 146        |
| 2.4 Список литературы .....  | 177        |
| <b>ПРИЛОЖЕНИЯ .....</b>  | <b>18</b>  |

## Описание программы

«Программирование на Python» на 2022-2023 уч. год

|  |  |
|--|--|
| Название программы                                   | <b>Программирование Python (14-18 лет)</b>   |
| Возраст обучающихся                                  | 14-18 лет  |
| Длительность программы (в часах)                     | 72 часа.   |
| Количество занятий в неделю                          | 2 академических часа в неделю: 1 занятие по 2 часа (академический час – 45 мин)  |
| Цель, задачи   | <p>Целью программы является формирование у обучающихся интереса к техническим видам творчества и развитие логического, технического мышления, создание условий для творческой самореализации личности обучающегося, посредством получения навыков работы с современными компьютерными системами автоматизированного проектирования.</p> <p>Цель достигается при выполнении образовательных, воспитательных, развивающих и профориентационных задач</p>   |
| Краткое описание программы                           | <p>Модуль 1 «Введение в программирование». В модуле рассматриваются основные типы данных, принципы и понятия объектно-ориентированного программирования</p> <p>Модуль 2 «Базовые конструкции»<br/>В модуле обучающиеся приобретают базовые навыки работы с основными конструкциями языка программирования; знакомство с комплексом базовых понятий и принципов функционального и объектно-ориентированного программирования (изучение структур данных, базовые принципы их обработки); получение опыта работы в интегрированной среде разработки на языке Python; формирование необходимых навыков работы с информацией (поиск, анализ, использование информации в сети Интернет). Дополнительно обучающиеся получат навыки работы с простой базой данных, а так же навыки создания запросов на сервисы в сети интернет</p> <p>Модуль 3 «Проекты Django, Задачи из ОГЭ и ЕГЭ» (для старшей группы).<br/>Модуль Django открывает группу проектных тем курса. В теме «Django» ученики рассматривают принципы создания веб-приложений на языке Python с использованием соответствующих библиотек. Тема подразумевает изучение платформы Django, разработка полноценных приложений и их поддержка. В теме «Задачи из ОГЭ и ЕГЭ» ученики научатся решать задачи по программированию из ГИА, на языке Python. Научаться создавать эффективные и оптимальные алгоритмы для решения задачи</p> |
| Первичные знания, необходимые для освоения программы | Базовые знания, полученные при изучении школьной программы информатики, математики и английского языка   |

|  |   |
|--|---|
| Результат освоения   | Выпускники будут иметь практические навыки программирования на языке Python, научатся решать задачи на программирование из ГИА, разрабатывать собственные веб-приложения на Django. Научаться работать в команде, разовьют личностные качества (активность, инициативность, волю, любознательность и т. п.). У ребят разовьется интеллект, внимание, память, восприятие, образное мышление и творческие способности; разовьются навыки анализа и оценки получаемой информации, разовьется мотивация к профессиональному самоопределению, сформируются навыки самоорганизации, воспитается самостоятельность, инициатива, творческая активность. |
| Перечень соревнований, в которых учащиеся смогут принять участие     | Конференция «Юные техники и изобретатели»<br>Соревнования Worldskills<br>Фестиваль идей и технологий «Rukami»<br>Научно-инженерная олимпиада «Кванториада»<br>Олимпиада НТИjunior   |
| Перечень основного оборудования, необходимого для освоения программы | Ноутбук, наушники, МФУ, моноблочное интерактивное устройство, напольная мобильная стойка для интерактивных досок или универсальное настенное крепление, доска магнитно-маркерная настенная, флипчарт магнитно-маркерный на треноге, сетевой фильтр  |
| Преимущества данной программы (отличия от других подобных курсов)    | В процессе программирования дети получат дополнительное образование в области математики и информатики, а также знания в области технического английского языка. Программа предполагает возможность участия обучающихся в соревнованиях, олимпиадах и конкурсах, таких как Программные решения для бизнеса JuniorSkills и WorldSkills   |

# **РАЗДЕЛ 1. КОМПЛЕКС ОСНОВНЫХ ХАРАКТЕРИСТИК ПРОГРАММЫ**

## **1.1 Пояснительная записка**

Современный мир невозможно представить без программирования и его продуктов во всём их многообразии. Автоматизация коренным образом изменила большую часть отраслей человеческой деятельности. Программисты, инженеры по автоматизации и системные администраторы неизменно присутствуют в списке самых востребованных специальностей на рынке образования и труда. В настоящее время мы переживаем большие изменения в развитии общества. В современную жизнь человека всё больше внедряются компьютеры и информационные технологии. Всё большее значение приобретает умение человека грамотно обращаться с компьютером, причём зачастую не на пользовательском уровне, а на уровне начинающего программиста.

Российское образование в этой области не уступает в качестве зарубежным конкурентам, но фундаментальные основы изучения информатики в школе не достаточны для конкурентоспособности подрастающего поколения. Ситуация на рынке программирования постоянно меняется. Если в начале 2000-х веб-разработчики думали, что лучше всего учить PHP и Ruby, то в 2020 году быстрее всего развивается и растет Python. Одна из причин популярности Python – более простое оформление, чем в других языках. Его проще изучить с нуля даже новичку в программировании. Это самый популярный язык общего назначения: он используется для машинного обучения, аналитики, разработке игр и в науке о данных.

Дополнительная общеобразовательная общеразвивающая программа «Программирование на Python» относится к технической направленности.

По степени освоения предполагается разделение программы на 3 уровня – ознакомительный и базовый и усложненный.

По форме организации содержания и процесса педагогической деятельности программа является интегрированной.

Программа разработана в соответствии со следующими нормативными документами:

Федеральный закон от 29 декабря 2012 г. № 273-ФЗ «Об образовании в Российской Федерации» (с последующими изменениями)

Государственная программа Российской Федерации «Развитие образования» на 2013 - 2020 годы / Постановление Правительства РФ от 15.04.2014 г.№ 295 (ред. от 27.04.2016)

Концепция развития дополнительного образования детей до 2030 года утвержденной Распоряжением Правительства РФ от 24 апреля 2015г. №729-р.

Порядок организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам /Приказ МинПросвещения РФ от 09.11.2018 г. № 19

Методические рекомендации по проектированию дополнительных общеразвивающих программ (включая разноуровневые программы) / Письмо Минобрнауки РФ от 18.11.2015 г. N 09-3242

СанПин 2.4.3648-20 «Санитарно-эпидемиологические требования к организации воспитания и обучения, отдыха и оздоровления детей и молодежи» / Постановление Главного государственного санитарного врача Российской Федерации от 28.09.2020 г. № 28

### **Актуальность программы**

Изучение основ программирования связано с развитием целого ряда умений и навыков (организация деятельности, ее планирование и т.д.), которые носят общеинтеллектуальный характер и формирование которых – одна из приоритетных задач в образовании.

Программа «Программирование на Python» позволяет посредством формирования начальных навыков программирования подготовить платформу для изучения в дальнейшем более сложных языков программирования. Данный учебный курс позволит обучающемуся

самостоятельно моделировать алгоритмические конструкции. Одним из средств достижения запланированных результатов является язык программирования «Python»

Особенности языка Python: язык очень легок в изучении; язык является объектно-ориентированным, поэтому его реализация является самой понятной; на Python можно написать любую программу; широкий спектр применения; Python работает во всех операционных системах, так как это интерпретируемый язык; Python распространяется свободно на условиях лицензии GNU 2.0. Данная лицензия разрешает бессрочно использовать Python на любом количестве компьютеров в любых целях без оформления каких-либо дополнительных документов. Поэтому целесообразно использовать этот язык при изучении программирования школьников. В связи с этим становится очевидным актуальность предлагаемой программы.

В рамках изучения программы обучающиеся будут сталкиваться с необходимостью самостоятельной работы над заданиями: обучающиеся учатся решать задачи без помощи преподавателя. Для этого в содержании курса фигурируют задания, в которых для решения задачи необходимо найти информацию в сети Интернет; может потребоваться устранение ошибки, которую не так просто обнаружить; условие сформулировано недостаточно прозрачно и обучающемуся необходимо самостоятельно формализовать его (или задать правильные вопросы преподавателю). Все эти знания, умения и практические навыки решения актуальных задач, полученные на занятиях, готовят обучающихся к самостоятельной проектно-исследовательской деятельности с применением современных технологий.

**Педагогическая целесообразность** программы выражена в подборе интерактивных и практико-ориентированных форм занятий, способствующих формированию основных компетенций (информационных, коммуникативных, компетенций личного развития и др.). Данная дополнительная общеразвивающая программа направлена на развитие логического и пространственного мышления слушателя, способствует раскрытию творческого потенциала личности, формированию усидчивости и трудолюбия, приобретению практических умений и навыков в области компьютерных технологий, способствует интеллектуальному развитию обучающегося. Так же целесообразность программы заключается в создании такой методики изучения современных технологий программирования, которая даст почву для самообразования и практической, исследовательской, самостоятельной научной деятельности. Обучение нацелено на раннее выявление и становление талантливых детей как через приобретение знаний и умений, так и через развитие творческих навыков посредством участия в творческих конкурсных состязаниях, популяризации науки, научной, изобретательской деятельности.

Программа «Программирование на Python» составлена в виде трёх модулей (Модуль 1 «Введение в программирование», Модуль 2 «Базовые конструкции языка Python», Модуль 3 «Проекты Django, Задачи из ОГЭ и ЕГЭ»), позволяющих получить обучающимся необходимый объём знаний вне зависимости от уровня подготовки и потребности.

Ознакомительный уровень предполагает использование и реализацию общедоступных и универсальных форм организации материала, минимальную сложность предлагаемого для освоения содержания программы.

Базовый уровень предполагает использование и реализацию таких форм организации материала, которые допускают освоение специализированных знаний и языка, гарантированно обеспечивают трансляцию общей и целостной картины в рамках содержательно-тематического направления программы.

**Отличительные особенности** программы заключаются в том, что программа не дублирует школьный курс информатики, а является его дополнением с профориентационными целями. Программа предполагает возможность участия обучающихся в соревнованиях, олимпиадах и конкурсах, таких как JuniorSkills и WorldSkills и др. Преимущество программы выражено в подборе интерактивных и практико-ориентированных форм занятий, способствующих формированию основных компетенций у обучающихся.

В программе предложен авторский подход в части структурирования учебного материала, определения последовательности его изучения, путей формирования системы знаний, умений и способов деятельности, развития, воспитания и социализации обучающихся. Программа предполагает возможность участия обучающихся в соревнованиях, олимпиадах и конкурсах, таких как JuniorSkills и WorldSkills и др.

**Адресат программы** – школьники, проявляющие интерес к информационным технологиям, и имеющие фундаментальные знания по математике, информатике и английскому языку.

**Возраст обучающихся**, участвующих в реализации данной программы – 14-18 лет.

На обучение принимаются все желающие, без предварительной подготовки, по заявлению родителей или лиц, их заменяющих. Набор в объединение производится по желанию обучающихся и их родителей.

Содержание программы учитывает возрастные и психологические особенности обучающихся 14-18 лет, которые определяют выбор форм проведения занятий. Обучающиеся этого возраста отличаются эмоциональностью, стремлением к активной практической деятельности, поэтому основной формой проведения занятий выбраны практические занятия. Обучающихся также увлекает совместная, коллективная деятельность, так как резко возрастает значение коллектива, общественного мнения, отношений со сверстниками, оценка поступков и действий обучающегося со стороны не только старших, но и сверстников. Обучающийся стремится завоевать в их глазах авторитет, занять достойное место в коллективе. Поэтому в программу включены практические занятия соревновательного характера, которые позволяют каждому проявить себя и найти своё место в детском коллективе.

Также следует отметить, что обучающиеся данной возрастной группы характеризуются такими психическими процессами, как изменение структуры личности и возникновение интереса к ней, развитие абстрактных форм мышления, становление более осознанного и целенаправленного характера деятельности, проявление стремления к самостоятельности и независимости, формирование самооценки. Эти процессы позволяют положить начало формированию начального профессионального самоопределения обучающихся.

**Форма обучения** – очная, с возможностью применения дистанционных технологий.

**Срок реализации и объем программы** определяется содержанием программы и составляет 1 год (72 академических часа).

**Режим занятий:** 2 академических часа в неделю.

1 раз - 2 часа (академический час – 45 мин.).

Через каждые 45 минут занятия следует 15-минутный перерыв, согласно требованиям СанПиН.

Количество обучающихся в группе 12 человек.

## 1.2 Цель и задачи программы

**Целью** программы является формирование творческой личности, обладающей информационными компетенциями, владеющей базовыми понятиями теории алгоритмов, умеющей разрабатывать эффективные алгоритмы и реализовывать их в виде программы, написанной на языке программирования Python.

**Задачи:**

Обучающие:

- познакомить с понятиями алгоритма, вычислимой функции, языка программирования;
- научить составлять и читать блок-схемы;
- сформировать навыки выполнения технологической цепочки разработки программ средствами языка программирования Python;

- объяснить основные конструкции языка программирования Python, позволяющие работать с простыми и составными типами данных (строками, списками, кортежами, словарями, множествами);
- научить применять функции при написании программ на языке программирования Python;
- научить отлаживать и тестировать программы, делать выводы о работе этих программ.

Метапредметные (развивающие):

- развивать познавательные процессы (внимание, восприятие, логическое мышление, память);
- развивать креативность;
- развивать способность к самореализации.

Личностные (воспитательные):

- воспитывать навыки самоорганизации;
- воспитывать навыки сотрудничества: работа в коллективе, в команде, микро-группе;
- воспитывать бережное отношение к технике, терпение в работе;
- воспитывать аккуратность, стремление доводить работу до конца;
- воспитывать самостоятельность, инициативу, творческую активность.

Здоровье-сберегающие:

- привить стремление к физическому развитию и укреплению здоровья обучающегося;
- сформировать правильную осанку;
- обеспечить эмоциональный комфорт и позитивное психологическое самочувствие обучающегося в процессе общения со сверстниками и взрослыми;
- формировать у обучающегося культуру сохранения и совершенствования собственного здоровья.

### 1.3 Календарный учебный график

| Этапы образовательного процесса                                     | 1 год обучения                                   |
|---|--|
| Начало учебного года  | 01.09.2022г.                                     |
| Окончание учебного года   | 25.05.2023г.                                     |
| Продолжительность учебного года (учебные часы)                      | 36 недель (72 часа)                              |
| Входной контроль знаний   | с 20.09.2022г. по<br>24.09.2022г.                |
| Текущий контроль успеваемости                                       | В течение всего периода освоения программы       |
| Промежуточная аттестация обучающихся                                | с 20.12.2022г. по<br>24.12.2022г.                |
| Итоговая аттестация   | с 20.05.2023г. по<br>24.05.2023г.                |
| Продолжительность учебных занятий                                   | 14-18 лет (45 минут)                             |
| Каникулы зимние   | 01 января – 08 января                            |
| Каникулы летние   | 01 июня – 31 августа                             |
| Дополнительные дни отдыха, связанные с государственными праздниками | 4 ноября, 31 декабря, 7 марта, 1-3 мая, 7-10 мая |

## 1.4 Учебно-тематический план

| №<br>пп/п | Название модуля,<br>темы  | Количество часов |           |           | Формы<br>аттестации/контроля   |
|-----------|---|------------------|-----------|-----------|--------------------------------|
|           |   | Всего            | Теория    | Практика  |                                |
| 1         | <b>Введение</b>   | 2                | 2         | -         | Входной:<br>Тестирование       |
| 2         | <b>Модуль 1.<br/>Ознакомительный.<br/>«Введение<br/>в программирование»</b> | <b>32</b>        | <b>10</b> | <b>22</b> |                                |
| 2.1       | Тема 1. Знакомство с PyCharm.Переменные.                                    | 4                | 2         | 2         | Текущий:<br>Наблюдение, беседа |
| 2.2       | Тема 2. Первые программы на языке Python, основные операторы                | 4                | 2         | 2         | Текущий:<br>Наблюдение, беседа |
| 2.3       | Тема 3. Условный оператор If.   | 2                | 1         | 1         | Текущий:<br>Наблюдение, беседа |
| 2.4       | Тема 4. Циклы в языке Python  | 4                | 2         | 2         | Текущий:<br>Наблюдение, беседа |
| 2.5       | Тема 5. Решение задач по изученным темам.                                   | 4                | -         | 4         | Текущий:<br>Наблюдение, беседа |
| 2.6       | Тема 6. Списки и кортежи в языке Python.                                    | 4                | 2         | 2         | Текущий:<br>Наблюдение, беседа |
| 2.7       | Тема 7. Работа с функциями в Python.  | 4                | 1         | 3         | Текущий:<br>Наблюдение, беседа |
| 2.8       | Тема 8. Решение задач по изученным темам.                                   | 4                | -         | 4         | Текущий:<br>Наблюдение, беседа |
| 2.9       | Промежуточная аттестация  | 2                | -         | 2         | Промежуточный:<br>Тестирование |
| 3         | <b>Модуль 2.<br/>«Базовые конструкции»</b>                                  | <b>10</b>        | <b>5</b>  | <b>5</b>  |                                |
| 3.1       | Тема 9. Основные понятия ООП  | 4                | 2         | 2         | Текущий:<br>Наблюдение, беседа |
| 3.2       | Тема 10. Парадигмы ООП  | 2                | 1         | 1         | Текущий:<br>Наблюдение, беседа |
| 3.3       | Тема 11. HTTP запросы на Python   | 4                | 2         | 2         | Текущий:<br>Наблюдение, беседа |
| 4         | <b>Модуль 3.<br/>«Проекты Django,<br/>Задачи из ОГЭ и ЕГЭ»</b>              | <b>20</b>        | <b>9</b>  | <b>11</b> |                                |
| 4.1       | Тема 12. Python в решении задач из ОГЭ                                      | 3                | 1         | 2         | Текущий:<br>Наблюдение, беседа |
|           | Тема 13. Python в решении задач из ЕГЭ                                      | 3                | 1         | 2         | Текущий:<br>Наблюдение, беседа |

|       |   |    |    |    |                                |
|-------|---|----|----|----|--------------------------------|
| 4.2   | Тема 14. Django. Понятия и первое приложение        | 2  | 1  | 1  | Текущий:<br>Наблюдение, беседа |
| 4.3   | Тема 15. Django. Шаблонизация и определение страниц | 2  | 1  | 1  | Текущий:<br>Наблюдение, беседа |
| 4.4   | Тема 16. Django. Модели объектов и миграции в БД    | 2  | 1  | 1  | Текущий:<br>Наблюдение, беседа |
| 4.5   | Тема 17. Django. Сложные древовидные модели         | 2  | 1  | 1  | Текущий:<br>Наблюдение, беседа |
| 4.6   | Тема 18. Django. Формы отправки                     | 2  | 1  | 1  | Текущий:<br>Наблюдение, беседа |
| 4.7   | Тема 19. Django. Администрирование приложения       | 2  | 1  | 1  | Текущий:<br>Наблюдение, беседа |
| 4.8   | Тема 20. Django. Авторизация и разделение прав      | 2  | 1  | 1  | Текущий:<br>Наблюдение, беседа |
| 5     | Подготовка проектов                                 | 4  | -  | 4  | Текущий:<br>Наблюдение, беседа |
| 6     | Итоговая аттестация                                 | 4  | -  | 4  | Итоговый: Защита проектов      |
| Итого |   | 72 | 26 | 46 |                                |

## 1.5 Содержание программы

### Введение

**Теоретическая часть:** Введение в образовательную программу. Ознакомление обучающихся с программой, приёмами и формами работы. Вводный инструктаж по ТБ.

### Модуль 1. Ознакомительный. «Введение в программирование»

#### Тема 1. Знакомство с PyCharm.Переменные.

**Теоретическая часть:** Знакомство со средой программирования на языке Python (PyCharm), изучение основных элементов интерфейса, запуск программ. Изучение понятия «переменная», «значение переменной».

**Практическая часть:** Работа со средой программирования PyCharm.

#### Тема 2. Первые программы на языке Python, основные операторы.

**Теоретическая часть:** Знакомство с операторами присваивания, ввода\вывода данных, разработка программ, реализующих линейные алгоритмы на языке программирования Python.

**Практическая часть:** Написание простых программ на языке программирования Python.

#### Тема 3. Условный оператор If.

**Теоретическая часть:** Рассмотрение условного оператора If, его области применения и особенности использования. Условные конструкции для разных программных ситуаций

**Практическая часть:** Разработка программ, реализующих условные алгоритмы. Тестирование программы с различными входными данными

#### Тема 4. Циклы в языке Python.

**Теоретическая часть:** Формат оператора цикла с предусловием while, оператора цикла с параметром for на языке программирования Python. Конструкции циклов с постусловием по средствам условного оператора.

**Практическая часть:** Разработка программ, циклические алгоритмы, имитация различных программных ситуаций с применением.

### **Тема 5. Решение задач по изученным темам.**

**Практическая часть:** Решение дополнительных задач по темам «Условный оператор If», «Циклы в языке Python».

### **Тема 6. Списки в языке Python.**

**Теоретическая часть:** Понятие «список» и «кортеж» в языке программирования Python. Сфера применения списков, способы «перебора» списков и кортежей с выводом информации

**Практическая часть:** Создание списка, различные способы задания списка, вывод элементов списка и кортежа на экран, основные функции по работе со списками и кортежами в языке программирования Python.

### **Тема 7. Работа с функциями в Python.**

**Теоретическая часть:** Вспомогательный алгоритм при разработке программ, понятие «функция» в языке программирования Python, описание функции, структура функции, обращение к функции в тексте программ.

**Практическая часть:** Приемы написания программ с использованием вспомогательных алгоритмов. Помещение функций и методов в отдельные исполнительные файлы и вызов их в основном исполнительном файле

### **Тема 8. Решение задач по изученным темам**

**Практическая часть:** Решение дополнительных задач по темам «Списки и кортежи», «Функции»

## **Модуль 2. «Базовые конструкции»**

### **Тема 9. Основные понятия ООП**

**Теоретическая часть:** Понятия объектно-ориентированного программирования, суть классов и объектов, использование функций в классах. Рассмотрение примеров из реальной жизни, визуальное построение класса

**Практическая часть:** Написание собственных классов и создание объектов. Помещение классов и их функций в отдельные исполнительные файлы и использование их в основном исполнительном файле

### **Тема 10. Парадигмы ООП**

**Теоретическая часть:** Разбор главных парадигм (столов) ООП, таких как наследование, полиморфизм, инкапсуляция. Изучение областей и способов применения парадигм ООП, а так же их комбинирования между собой

**Практическая часть:** Применение парадигм ООП к ранее созданным классам. Комбинирование парадигм в классах и их методах

### **Тема 11. HTTP запросы на Python**

**Теоретическая часть:** Изучение понятий HTTP Request, API, методов GET и POST, рассмотрение примеров запросов, изучение схем взаимодействия между API и сервером

**Практическая часть:** Создание различных запросов на API сервисы, запросы с GET параметрами, запросы с JSON параметрами внутри для POST запросов

## **Модуль 3. «Проекты Django, Задачи из ОГЭ и ЕГЭ»**

### **Тема 12. Python в решении задач из ОГЭ**

**Теоретическая часть:** Разбор задач из ОГЭ, определение эффективных методов решения. Адаптация методов решения исходя из разных типов задач

**Практическая часть:** Решение задач из ЕГЭ по средствам языка Python, использование эффективных методов решения

## **Тема 13. Python в решении задач из ЕГЭ**

*Теоретическая часть:* Разбор задач из ЕГЭ, определение эффективных методов решения. Адаптация методов решения исходя из разных типов задач

*Практическая часть:* Решение задач из ЕГЭ по средствам языка Python, использование эффективных методов решения

## **Тема 14. Django. Понятия и первое приложение**

*Теоретическая часть:* Понятия фреймворка, веб-приложений и HTTP серверов, рассмотрение структуры программы, изучение схемы взаимодействий внутри фреймворка. Разные варианты инициализации фреймворка и другие варианты сервера для приложения

*Практическая часть:* Создание и настройка своего веб-приложения, создание главной страницы, размещение контента на странице

## **Тема 15. Django. Шаблонизация и определение страниц**

*Теоретическая часть:* Изучение шаблонизатора, его инструментов и способов применения, способ определения перехода по страницам, вызовы различных действий при переходе на определенную страницу

*Практическая часть:* Подключение шаблонизатора к приложению, вывод контента через вызов функций шаблонизатора, указание определения переходов по страницам и вызов действий при переходе на страницу

## **Тема 16. Django. Модели объектов и миграции в БД**

*Теоретическая часть:* Понятие модели объекта, его принцип работы, изучение миграции свойств и параметров в базу данных. Рассмотрение возможных вариантов создания модели, внутренних компонентов и метаданных модели

*Практическая часть:* Создание собственных моделей объектов и миграции их в базу данных, описание модели через метаданные, создание полей в модели

## **Тема 17. Django. Сложные древовидные модели**

*Теоретическая часть:* Определение иерархической модели и какие способы их построения есть. Рассмотрение классических вариантов иерархии объектов, алгоритмов создания древовидной иерархии. Изучение библиотеки Django MPTT для создания иерархических моделей

*Практическая часть:* Создание древовидной модели с выводом её из базы данных на страницу. Оформление древовидной иерархии в веб-интерфейсе с применением языков CSS и HTML

## **Тема 18. Django. Формы отправки данных**

*Теоретическая часть:* Что такое формы в веб интерфейсе, какие данные можно отправить и как их проверять. Варианты разметки формы, определение полей под пароли и логины с особенностями их отображения

*Практическая часть:* Размещение формы на странице, её оформление, проверка и отправка данных в функцию для обработки данных, отправка проверенных данных в базу данных

## **Тема 19. Django. Администрирование приложения**

*Теоретическая часть:* Рассмотрение приложения администрирования сайта, её кастомизации, варианты использования и способов взаимодействия с ней. Способы подключения моделей в панель администратора

*Практическая часть:* Подключение и настройка приложения администрирования, создание суперпользователя для администрирования сайта, подключение моделей в панель администратора, настройка модели и её отображения

## **Тема 20. Django. Авторизация и разделение прав**

**Теоретическая часть:** Определение понятия авторизации в системе, рассмотрение возможных прав доступа, виды пользователей в системе. Понятие сессии и как ограничить права пользователя в рамках сессии

**Практическая часть:** Создание и подключение формы для авторизации пользователя, организация пользователей в панели администрирования, разделение пользователей по правам доступа и их возможностям

### **Подготовка проектов.**

**Практическая часть:** Разработка индивидуального или группового проекта на языке программирования Python.

### **Итоговая аттестация**

**Практическая часть:** Защита индивидуальных или групповых проектов.

## **1.6. Планируемые результаты**

В процессе занятий по программе к окончанию учебного года обучающиеся будут знать:

- основные конструкции и идиомы языка программирования Python;
- термины «информация», «сообщение», «данные», «алгоритм», «программа»; понимание различий между употреблением этих терминов в обыденной речи и в информатике;
- логические значения, операции и выражения с ними;
- понятия класс, объект, обработка событий;
- методы проектной деятельности.

В процессе занятий по программе к окончанию учебного года обучающиеся будут уметь:

- на практике составить несложную программу для выполнения поставленной аналитической задачи;
- составлять линейные, разветвляющиеся и циклические алгоритмы управления исполнителями на языке программирования Python;
- использовать готовые прикладные компьютерные программы и сервисы;
- выбирать способы представления данных в зависимости от поставленной задачи;
- создавать мини-проекты на основе полученных знаний.

По итогам освоения программы, к окончанию учебного года, обучающиеся приобретут:

- *Метапредметные результаты:*

- формирование навыков самоорганизации;
- формирование навыков сотрудничества: работа в коллективе, в команде, микрогруппе;
- воспитание бережного отношения к технике;
- воспитание самостоятельности, инициативности;
- развитие навыков анализа и оценки получаемой информации.

- *Личностные:*

- развитие личностных качеств (активность, инициативность, воля, любознательность и т. п.);
- развитие внимания, памяти, восприятия, образного мышления;
- развитие логического и пространственного воображения;
- развитие творческих способностей и фантазии;
- развитие мотивации к познанию и творчеству;
- формирование положительных черт характера: трудолюбия, аккуратности, собранности, усидчивости, отзывчивости;
- развитие мотивации к профессиональному самоопределению.

## РАЗДЕЛ 2. КОМПЛЕКС ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИХ УСЛОВИЙ

### 2.1 Условия реализации программы

#### **Материально-техническое обеспечение:**

Занятия проходят в помещении с оптимальными условиями, отвечающими требованиям СанПиН, на базе Центра цифрового образования детей «IT-куб» г. Сатка.

Для реализации учебных занятий используется следующее оборудование и материалы:

#### **Оборудование и расходные материалы:**

- компьютеры и ноутбуки на каждого обучающегося и преподавателя;
- принтер;
- проекционное оборудование (экраны);
- магнитно-маркерная доска;
- интерактивная панель;
- whiteboard маркеры;
- бумага писчая;
- шариковые ручки;
- permanent маркеры.

#### **Информационное обеспечение:**

- операционная система на базе Linux

#### **Методическое обеспечение:**

- варианты демонстрационных программ, материалы по терминологии ПО;
- инструкции по настройке оборудования;
- учебная и техническая литература;
- набор цифровых образовательных ресурсов – дидактические материалы, интерактивные тесты, анимационные плакаты.

### 2.2 Формы аттестации обучающихся

Система контроля знаний и умений обучающихся представляется в виде:

*Входящий контроль* осуществляется при комплектовании группы в начале учебного

года.

Цель – определить исходный уровень знаний обучающихся, определить формы и методы работы с обучающимися.

Форма контроля: тестирование.

*Текущий контроль* осуществляется после изучения отдельных тем, раздела программы.

В практической деятельности результативность оценивается качеством выполнения

практических работ, поиску и отбору необходимого материала, умению работать с различными источниками информации. Анализируются положительные и отрицательные стороны работы, корректируются недостатки. Контроль знаний осуществляется с помощью заданий педагога (решение практических задач средствами языка программирования); взаимоконтроля, самоконтроля и др. Они активизируют, стимулируют работу обучающихся, позволяют более полно проявлять полученные знания, умения, навыки.

*Промежуточная аттестация* осуществляется в конце I полугодия учебного года.

Форма контроля: тестирование, решение практических задач средствами языка программирования.

*Итоговая аттестация* осуществляется в конце учебного года.

Форма контроля: защита проекта.

Проект является одним из видов самостоятельной работы, предусмотренной в ходе обучения по программе. Педагог-наставник оказывает консультационную помощь в выполнении проекта.

Индивидуальный (групповой) проект оценивается формируемой комиссией. Состав комиссии (не менее 3-х человек): педагог-наставник, администрация учебной организации, приветствуется привлечение ИТ-профессионалов, представителей высших и других учебных заведений.

Компонентами оценки индивидуального (группового) проекта являются (по мере убывания значимости): качество индивидуального проекта, отзыв руководителя проекта, уровень презентации и защиты проекта. Если проект выполнен группой обучающихся, то при оценивании учитывается не только уровень исполнения проекта в целом, но и личный вклад каждого из авторов. Решение принимается коллегиально.

### **Формы организации учебного занятия по программе**

В образовательном процессе помимо традиционного учебного занятия используются многообразные формы, которые несут учебную нагрузку и могут использоваться как активные способы освоения детьми образовательной программы, в соответствии с возрастом обучающихся, составом группы, содержанием учебного модуля:

- беседа;
- лекция;
- мастер-класс;
- практическое занятие;
- защита проектов;
- конкурс;
- викторина;
- диспут;
- круглый стол;

Некоторые формы проведения занятий могут объединять несколько учебных групп или весь состав объединения, например, экскурсия, викторина, конкурс и т. д.

В данной программе применяются следующие педагогические технологии:

- технология индивидуализации обучения;
- технология группового обучения;
- технология коллективного взаимообучения;
- технология дифференцированного обучения;
- технология разноуровневого обучения;
- технология проблемного обучения;
- технология развивающего обучения;
- технология дистанционного обучения;
- технология игровой деятельности;
- коммуникативная технология обучения;
- технология коллективной творческой деятельности;

- технология решения изобретательских задач;
- здоровье-сберегающая технология.

Методическое обеспечение учебного процесса включает разработку преподавателем методических пособий, вариантов демонстрационных программ и справочного материала.

### 2.3 Оценочные и методические материалы

В программу входят разнообразные оценочные материалы, в зависимости от темы занятия (Приложение).

Организация образовательного процесса в данной программе происходит в очной форме обучения, с возможностью применения дистанционных технологий, и групповой форме.

При реализации программы используются различные методы обучения:

- объяснительно-иллюстративный (предъявление информации различными способами (объяснение, рассказ, беседа, инструктаж, демонстрация, работа с технологическими картами и др.);
- проблемный (постановка проблемы и самостоятельный поиск её решения обучающимися);
- репродуктивный (воспроизведение знаний и способов деятельности по аналогу);
- поисковый (самостоятельное решение проблем);
- метод проблемного изложения (постановка проблемы педагогам, решение ее самим педагогом, соучастие обучающихся при решении);
- метод проектов (технология организации образовательных ситуаций, в которых обучающийся ставит и решает собственные задачи).

Для оценки результативности обучения и воспитания регулярно используются разнообразные методы: наблюдение за деятельностью; метод экспертной оценки преподавателем, мотивация, убеждение, поощрение, упражнение, стимулирование, создание ситуации успеха. Данные методы используются при анализе деятельности обучающихся, при организации текущей, промежуточной и итоговой аттестации обучающихся.

Перечисленные выше методы обучения используются в комплексе, в зависимости от поставленных целей и задач.

## 2.4 Список литературы

### Список литературы для педагога:

1. Бонцанини М. Анализ социальных медиа на Python. Извлекайте и анализируйте данные из всех уголков социальной паутины на Python - Издательство "ДМК Пресс" - 2018 - ISBN: 978-5-97060-574-5 - Текст электронный // ЭБС Лань - URL: <https://e.lanbook.com/book/108129>
2. Основы алгоритмизации и программирования на Python : учеб. пособие / С.Р. Гуриков. - М. : ФОРУМ : ИНФРА-М, 2018. - 343 с. - (Среднее профессиональное образование-Режим доступа:
3. <http://znanium.com/catalog/product/961522>
4. Язык программирования Python: практикум : учеб. пособие / Р.А. Жуков- М. : ИНФРА-М, 2019. - 216 с. + Доп. материалы [Электронный ресурс; Режим доступа: <http://www.znanium.com>]. - (Высшее образование: Бакалавриат). - [ww.dx.doi.org/10.12737/textbook\\_5cb5ca35aaa7f5.89424805](http://dx.doi.org/10.12737/textbook_5cb5ca35aaa7f5.89424805).

### Список литературы для обучающихся:

1. Златопольский Д.М. Основы программирования на языке Python. – М.: ДМК Пресс,2017.–284с.ил."
2. Тарасов Д. Факультативный курс «Основы алгоритмизации и программирования на языке Python» 10-11 классы. Видеоуроки для учителей, родителей и учеников.

### Электронные ресурсы:

1. <https://nuancesprog.ru/p/1154/#i>
2. <https://pythonworld.ru/samouchitel-python>

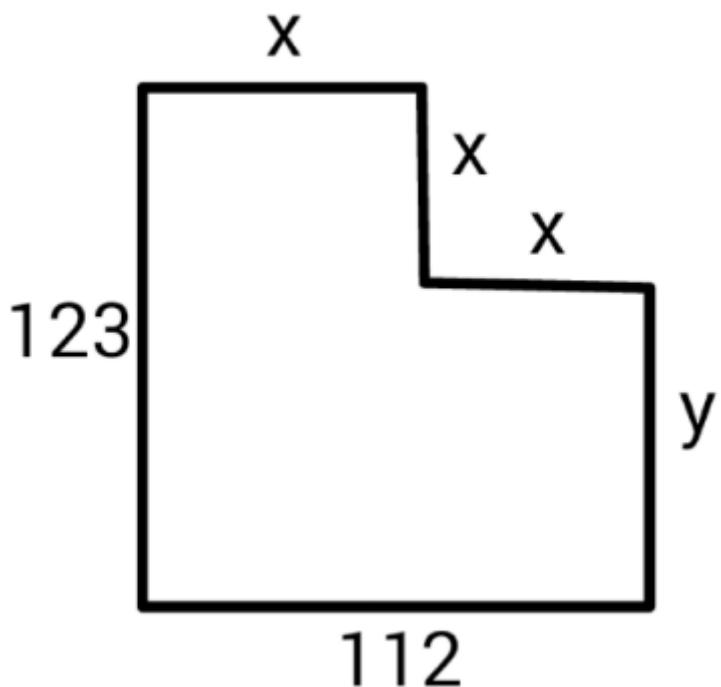
## ПРИЛОЖЕНИЯ

### Приложение 1. Входной контроль.

Для обучающихся – 12-17 лет

Фамилия, Имя

Из прямоугольника вырезали квадрат со стороной  $x$ , получилась фигура как на рисунке.  
Чему равна сумма цифр?



В алфавите племени мумба-юмба 32 буквы. Любое слово в языке этого племени состоит из пяти букв и должно одинаково читаться справа налево и слева направо, при этом первые две буквы слова обязательно различаются, а третья совпадает с пятой.

Каково максимальное количество слов в этом языке?

Катя наклеила на рулет тонкие поперечные кольца трёх разных цветов. Если разрезать по серым кольцам, получится 25 кусков рулета, если по малиновым – 47 кусков, а если по зеленым – 31 кусок.

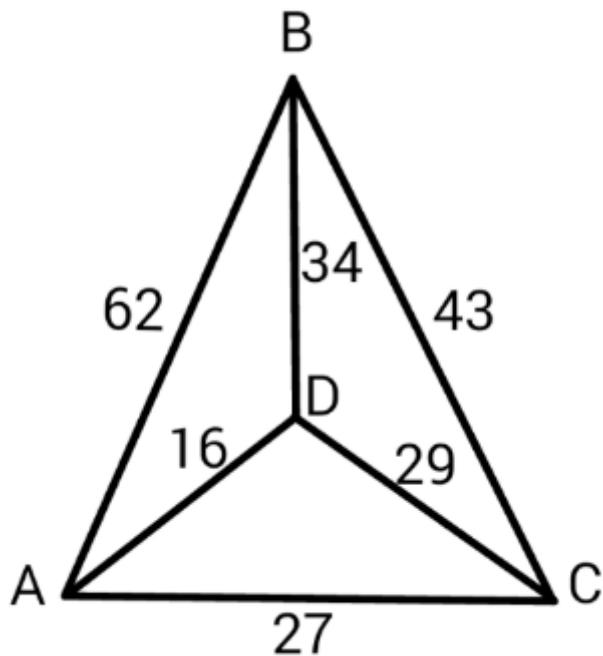
Сколько кусков рулета получится, если разрезать по кольцам всех трёх цветов?

#### Примечания

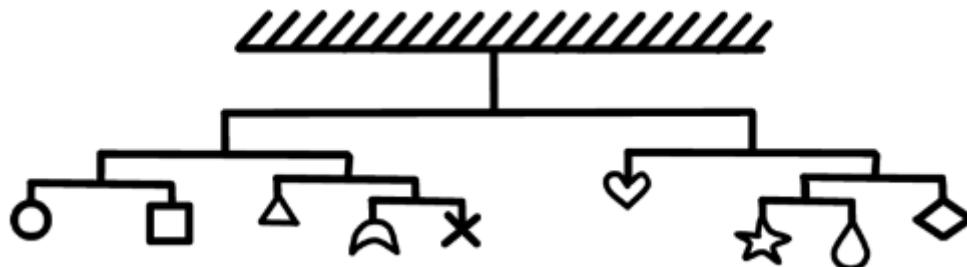
Учтите, что наклеить кольцо одного цвета на кольцо другого нельзя.

На рисунке показано расположение городов А, В, С и D и расстояния между ними. Турист выходит из города В и собирается посетить остальные города, побывав в каждом по разу.

Какова наименьшая возможная длина маршрута, если он хочет закончить свой путь в том же городе?



Фигурки, общей массой 432 грамма, при помощи невесомых нитей и планок собрали в конструкцию, изображённую на рисунке. Оказалось, что все её части находятся в равновесии. Сколько весит ромб?



Люди переезжают в города, за год численность людей удваивается. Если люди заселят весь город за 12 лет, то сколько лет понадобится, чтобы занять лишь половину города?

Сравните пары слов. Сколько среди них полностью идентичных?

O/Sanmarco  
Φ.Wagonerrte  
A.S.Schmetterling  
N.V.Murfreesboroque  
P.S.Splendoursec

O/Samnarco  
Φ.Wagonertre  
A.S.Schnetterling  
N.V.Munfreesboroque  
P.S.Sqlendoursec

Семь человек выясняли, какой сегодня день недели.

Первый сказал: «Послезавтра – воскресенье».

Второй: «Вчера был понедельник».

Третий: «Завтра будет суббота».

Четвертый: «Завтра будет среда».

Пятый: «Вчера был четверг».

Шестой: «Позавчера было воскресенье».

Седьмой: «Позавчера была среда».

Какой сегодня день недели, если трое ошибаются?

Вам предложены несколько высказываний и следствие из них (выделено жирным).

Согласны ли Вы с этим следствием?

Все клёны — растения.

Некоторые растения быстро желтеют.

**Значит, некоторые клёны быстро желтеют.**

Да

Ннет

Гусеница прогрызает яблоко диаметром 6 сантиметров насеквоздь за 16 секунд, вылезая снаружи полностью.

Известно, что середину яблока она начинает грызть уже через 6 секунд после начала пути.

Какова длина гусеницы в сантиметрах?

### Для обучающихся 11-13 лет

**Фамилия, Имя**

Назовите два числа, у которых количество цифр равно количеству букв, составляющих название каждого из этих чисел.

Собака была привязана к десятиметровой веревке, а прошла двести метров. Как ей это удалось?

Есть дорога, по которой может проехать только одна машина. По дороге едут две машины: одна с горы, другая под гору. Как им разъехаться?

Как с помощью только одной палочки образовать на столе треугольник?

В каком месяце болтливая девочка говорит меньше всего?

Что становится на треть больше, если его поставить вверх ногами?

Представьте себе, что вы кондуктор. Поезд везет сто вагонов, в каждом вагоне 10 купе, в каждом купе 4 пассажира. Сколько лет кондуктору?

Юра разрезал огромную пиццу на 10 кусков. Затем он взял один из кусков и разрезал его еще на 10. После этого из имеющихся кусков он выбрал два и разрезал каждый из них на 10.

**Вопрос:** Сколько кусков пиццы получилось у Юры.

В компьютерной игре нужно победить монстра. Изначально у Юры было только 9 выстрелов. Но за каждое попадание он получал дополнительно еще 3 выстрела.

**Вопрос:** Сколько раз Юра попал в цель, если всего он выстрелил 30 раз, израсходовав все выстрелы?

#### Критерии оценивания:

Низкий уровень: 0-70% выполненных заданий;

Средний уровень: 70-85% правильно выполненных заданий;

Высокий уровень: 85-100% правильно выполненных заданий.

#### Приложение 2. Вопросы для промежуточного контроля по усвоению материала

Составьте выражение для вычисления в интерпретаторе Python 3 и вставьте в поле ответа результат вычисления:

11111·1111111

— произведение чисел 11111 (5 единиц) и 1111111 (7 единиц)

Запишите число 1.2345e3 в виде [десятичной дроби](#).

Составьте и запишите выражение для вычисления:

$2014.0^{14}$ (Возвестив 14 степень)

Обратите внимание на запись числа: это вещественное число.

Приведите к целому типу число 2.99

Расставьте скобки в выражении

a and b or not a and not b

в соответствии с порядком вычисления выражения (приоритетом операций). Всего потребуется 5 пар скобок (внешние скобки входят в их число).

Найдите результат выражения для заданных значений a и b

Учитывайте регистр символов при ответе.

|                            |   |       |
|----------------------------|---|-------|
| a                          | = | True  |
| b                          | = | False |
| a and b or not a and not b |   |       |

Отметьте выражения, значения которых равны True:

- "239" < "30" and 239 < 30
- "239" < "30" and 239 > 30
- "239" > "30" and 239 < 30
- "239" > "30" and 239 > 30

Укажите результат выражения:

"123" + "42"

Какое значение будет у переменной i после выполнения фрагмента программы?

```
i = 0
while i <= 10:
    i = i + 1
    if i > 7:
        i = i + 2
```

Сколько итераций цикла будет выполнено в этом фрагменте программы?

```
i = 0
while i <= 10:
    i = i + 1
    if i > 7:
        i = i + 2
```

Сколько всего знаков \* будет выведено после исполнения фрагмента программы:

```
i = 0
while i < 5:
    print('*')
    if i % 2 == 0:
        print('**')
    if i > 2:
        print('***')
    i = i + 1
```

Определите, какое значение будет иметь переменная i после выполнения следующего фрагмента программы:

```
i = 0
s = 0
while i < 10:
    i = i + 1
    s = s + i
    if s > 15:
        break
    i = i + 1
```

Определите, какое значение будет иметь переменная i после выполнения следующего фрагмента программы:

```
i = 0
s = 0
```

```
while i < 10:  
    i = i + 1  
    s = s + i  
    if s > 15:  
        continue  
    i = i + 1
```

### Критерии оценивания:

Низкий уровень: 0-70% выполненных заданий;

Средний уровень: 70-85% правильно выполненных заданий;

Высокий уровень: 85-100% правильно выполненных заданий.

### Приложение 3. Задачи для промежуточного контроля по усвоению материала

Напишите простой калькулятор, который считывает с пользовательского ввода три строки: первое число, второе число и операцию, после чего применяет операцию к введённым числам ("первое число" "операция" "второе число") и выводит результат на экран.

Поддерживаемые операции: +, -, /, \*, mod, pow, div, где  
mod - это взятие остатка от деления,  
pow - возведение в степень,  
div - целочисленное деление.

Если выполняется деление и второе число равно 0, необходимо выводить строку "Деление на 0!".

Обратите внимание, что на вход программе приходят вещественные числа.

Жители страны Малевии часто экспериментируют с планировкой комнат. Комнаты бывают треугольные, прямоугольные и круглые. Чтобы быстро вычислять жилплощадь, требуется написать программу, на вход которой подаётся тип фигуры комнаты и соответствующие параметры, которая бы выводила площадь получившейся комнаты. Для числа  $\pi$  в стране Малевии используют значение 3.14.

Формат ввода, который используют Малевийцы:

Треугольник

a

b

c

где a, b и c — длины сторон треугольника

прямоугольник

a

b

где a и b — длины сторон прямоугольника

круг

r

где r — радиус окружности

Напишите программу, которая получает на вход три целых числа, по одному числу в строке, и выводит на консоль в три строки сначала максимальное, потом минимальное, после чего оставшееся число.

На ввод могут подаваться и повторяющиеся числа.

Паша очень любит кататься на общественном транспорте, а получая билет, сразу проверяет, счастливый ли ему попался. Билет считается счастливым, если сумма первых трех цифр совпадает с суммой последних трех цифр номера билета.

Однако Паша очень плохо считает в уме, поэтому попросил вас написать программу, которая проверит равенство сумм и выведет "Счастливый", если суммы совпадают, и "Обычный", если суммы различны.

На вход программе подаётся строка из шести цифр. (**Пример:123321**)

Выводить нужно только слово "Счастливый" или "Обычный", с большой буквы.

Напишите программу, которая считывает с консоли числа (по одному в строке) до тех пор, пока сумма введённых чисел не будет равна 0 и **сразу после этого** выводит сумму квадратов всех считанных чисел.

Гарантируется, что в какой-то момент сумма введённых чисел окажется равной 0, **после этого считывание продолжать не нужно**.

В примере мы считываем числа 1, -3, 5, -6, -10, 13; в этот момент замечаем, что сумма этих чисел равна нулю и выводим сумму их квадратов, не обращая внимания на то, что остались ещё не прочитанные значения.

Напишите программу, которая выводит часть последовательности 1 2 2 3 3 3 4 4 4 4 5 5 5 5 5 ... (число повторяется столько раз, сколько равно). На вход программе передаётся неотрицательное целое число  $n$  — столько элементов последовательности должна отобразить программа. На выходе ожидается последовательность чисел, записанных через пробел в одну строку.

Например, если  $n = 7$ , то программа должна вывести 1 2 2 3 3 3 4.

Выведите таблицу размером  $n \times n$ , заполненную числами от 1 до  $n^2$  по спирали, выходящей из левого верхнего угла и закрученной по часовой стрелке, как показано в примере ([здесь  \$n=5\$](#) )

#### Критерии оценивания:

Низкий уровень: 0-70% выполненных заданий;

Средний уровень: 70-85% правильно выполненных заданий;

Высокий уровень: 85-100% правильно выполненных заданий.

#### Приложение 4. Итоговые результаты по программе Диагностическая карта

| № | ФИО | Входная диагностика | Промежуточная диагностика | Итоговая диагностика |
|---|-----|---------------------|---------------------------|----------------------|
|   |     |                     |                           |                      |
|   |     |                     |                           |                      |
|   |     |                     |                           |                      |
|   |     |                     |                           |                      |

Результативность отслеживается с помощью карты наблюдений, анализа участия детей в совместной продуктивной деятельности, разработки и защите творческих проектов. Усвоение программы возможно по 3-м уровням: низкий (Н), средний (С), высокий (В).

#### Низкий уровень

Обучающиеся должны знать / понимать:

технику безопасного поведения во время занятий; правила поведения в общественных местах; понятие программы Python; общую структуру программы; основные типы данных; оператор присваивания; назначение условного оператора; способ записи условного оператора; логический тип данных; логические операторы or, and, not; основные циклы с условием; основные правила записи циклов условием; формат записи цикла с параметром; понятие функции; основные принципы структурного программирования; понятие локальных переменных подпрограмм; способ передачи параметров; назначение строкового типа данных; операторы для работы со строками; операции со строками; способ описания списка; способ описания кортежа; способ описания словаря; основные операции, выполняемые со списками, кортежами и словарями; понятие множества; способы описания множества; операторы работы с множествами; что такое стиль программирования; правила именования объектов; основные рекомендации при написании программ.

*Обучающиеся должны уметь:*

понимать учебную задачу, сохранять ее содержание в процессе ее выполнения под руководством педагога; работать в паре, малой группе; выполнить установку программы под руководством педагога; выполнить простейшую программу в интерактивной среде; написать комментарии в программе под руководством педагога; решать задачи на элементарные действия с числами; использовать условный оператор; определять вид цикла, наиболее удобный для решения поставленной задачи; использовать цикл с условием под руководством педагога; определять целесообразность применения и использования цикла с параметром для решения поставленной задачи по наводящим вопросам педагога; создавать и использовать основные функции; описывать и соединять строки; находить подстроку в строке с помощью педагога; находить количество слов в строке; вводить и выводить элементы списка под руководством педагога; приводить примеры использования вложенных списков (матриц) по наводящим вопросам педагога; описывать множества под руководством педагога; определять принадлежность элемента множеству по наводящим вопросам; определять вид ошибок и находить ошибки в программе под руководством педагога; составлять элементарные алгоритмы для решения задач; реализовывать алгоритмы на компьютере в виде программ, написанных на языке Python под руководством педагога.

### **Средний уровень**

*Обучающиеся должны знать / понимать:*

основные положения техники безопасности на занятиях, правила поведения в общественных местах, правила дорожной безопасности; основные приемы взаимодействия в группе сверстников; понятие программы; структуру программы на Python; режимы работы с Python; типы данных; целые, вещественные типы данных и операции над ними; оператор присваивания; назначение условного оператора; способ записи условного оператора; логический тип данных; логические операторы or, and, not; циклы с условием и их виды; назначение и особенности использования цикла с параметром; формат записи цикла с параметром; примеры использования циклов различных типов; понятие функции; основные способы описания функции; принципы структурного программирования; понятие локальных переменных подпрограмм; понятие формальных и фактических параметров подпрограмм; способ передачи параметров; назначение строкового типа данных; операторы для работы со строками; процедуры и функции для работы со строками; операции со строками; сложные типы данных; способ описания списка; способ доступа к элементам списка; способ описания кортежа; способ описания словаря; операции, выполняемые со списками, кортежами и словарями; понятие множества; способы описания множества; операторы работы с множествами; что такое стиль программирования; правила именования объектов; основные рекомендации при написании программ; основные шаги работы над проектом, его презентации.

*Обучающиеся должны уметь:*

уважительно относится к преподавателям и сверстникам; применять некоторые приемы логического (абстрактное) мышления; концентрировать внимание на одном или двух объектах; понимать причины успеха/неуспеха с помощью анализа педагога; выполнить установку программы; выполнить простейшую программу в интерактивной среде; написать комментарии в программе; решать задачи на элементарные действия с числами; использовать условный оператор; создавать сложные условия с помощью логических операторов; определять вид цикла, наиболее удобный для решения поставленной задачи; использовать цикл с условием; определять целесообразность применения и использования цикла с параметром для решения поставленной задачи; создавать и использовать функции; использовать механизм параметров для передачи значений; описывать строки; соединять строки; находить длину строки; вырезать часть строки; находить подстроку в строке; находить количество слов в строке; описывать списки; вводить элементы списка; выводить элементы списка; выполнять поиск элемента в списке, поиск минимума и максимума, нахождение суммы элементов списка; использовать вложенные списки; приводить примеры использования вложенных списков (матриц); описывать множества; определять принадлежность элемента множеству; вводить элементы множества; выводить элементы множества; определять вид ошибок и находить ошибки в программе; составлять алгоритмы для решения задач; реализовывать алгоритмы на компьютере в виде программ, написанных на языке Python; отлаживать и тестировать программы, написанные на языке Python; понимать и сохранять цели и задачи учебной деятельности в ходе занятия; планировать свою деятельность с помощью взрослого; сотрудничать с взрослыми и сверстниками в процессе выполнения учебной задачи; понимать причины успеха/неуспеха учебной деятельности под руководством взрослого; делать выводы в ходе полученных заданий; выстраивать логические цепи рассуждений под руководством педагога; выражать творческие идеи, разработать творческий проект на основе образца; конструктивно взаимодействовать в составе группы в ходе работы над проектом.

### **Высокий уровень**

*Обучающиеся должны знать / понимать:*

основные положения техники безопасности на занятиях, правила поведения в общественных местах, правила дорожной безопасности, правила поведения во время чрезвычайных происшествий; приемы конструктивного взаимодействия в группе сверстников; понятие программы; структуру программы на Python; режимы работы с Python; общую структуру программы; типы данных; целые, вещественные типы данных и операции над ними; оператор присваивания; назначение условного оператора; способ записи условного оператора; логический тип данных; логические операторы or, and, not; циклы с условием и их виды; правила записи циклов условием; назначение и особенности использования цикла с параметром; формат записи цикла с параметром; примеры использования циклов различных типов; понятие функции; способы описания функции; принципы структурного программирования; понятие локальных переменных подпрограмм; понятие формальных и фактических параметров подпрограмм; способ передачи параметров; назначение строкового типа данных; операторы для работы со строками; процедуры и функции для работы со строками; операции со строками; сложные типы данных; способ описания списка; способ доступа к элементам списка; способ описания кортежа; способ описания словаря; операции, выполняемые со списками, кортежами и словарями; понятие множества; способы описания множества; операторы работы с множествами; что такое стиль программирования; правила именования объектов; основные рекомендации при написании программ; правила и этапы работы над проектом; приемы успешной презентации проекта.

*Обучающиеся должны уметь:*

выполнить установку программы; выполнить простейшую программу в интерактивной среде; написать комментарии в программе; решать задачи на элементарные действия с числами; использовать условный оператор; создавать сложные условия с помощью

логических операторов; определять вид цикла, наиболее удобный для решения поставленной задачи; использовать цикл с условием; определять целесообразность применения и использования цикла с параметром для решения поставленной задачи; создавать и использовать функции; использовать механизм параметров для передачи значений; описывать строки; соединять строки; находить длину строки; вырезать часть строки; находить подстроку в строке; находить количество слов в строке; описывать списки; вводить элементы списка; выводить элементы списка; выполнять поиск элемента в списке, поиск минимума и максимума, нахождение суммы элементов списка; использовать вложенные списки; приводить примеры использования вложенных списков (матриц); описывать множества; определять принадлежность элемента множеству; вводить элементы множества; выводить элементы множества; определять вид ошибок и находить ошибки в программе; составлять алгоритмы для решения задач; реализовывать алгоритмы на компьютере в виде программ, написанных на языке Python; отлаживать и тестировать программы, написанные на языке Python; понимать учебную задачу, анализировать достижение результата; делать выводы в ходе полученных заданий; самостоятельно выстраивать логические цепи рассуждений; понимать причины успеха/неуспеха, конструктивно принимать причины неуспеха, исправлять ошибки с минимальным участием педагога; планировать свою деятельность, находить оригинальные способы выполнения поставленной творческой задачи; создать проектировочную команду и организовать ее деятельность; разработать творческий проект по предложенной или самостоятельно выбранной теме в составе творческой группы; самостоятельно подготовить презентацию и защитить проект.