

СМОЛЕНСКОЕ ОБЛАСТНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ С ИНТЕРНАТОМ
«ЛИЦЕЙ ИМЕНИ КИРИЛЛА И МЕФОДИЯ»

МЕТОДИЧЕСКАЯ РАЗРАБОТКА

Дополнительная образовательная программа

Программирование на Python: первые шаги

Автор-составитель:
Сенчилова Ольга Константиновна,
учитель информатики.

Смоленск, 2023

Аннотация

Методическая разработка посвящена теме формирования у обучающихся интереса к информационным технологиям через изучение программирования, направлена на реализацию ФГОС и развитие интереса к техническим специальностям. Реализация этого направления представлена в формате дополнительной образовательной программы «Программирование на Python: первые шаги». Особенностью программы является использование компетентностного подхода: большинство заданий имеют сюжетный характер, формулируются в виде жизненных ситуаций, а изучение каждого модуля завершается созданием консольного или оконного приложения игрового характера.

Содержание

Аннотация. Содержание	2
1. Введение.....	3
2. Описание дополнительной образовательной программы «Программирование на Python: первые шаги»	4
3. Заключение	7
Приложения	8
Приложение 1. Дополнительная образовательная программа «Программирование на Python: первые шаги».....	8
Приложение 2. Комплексные ситуативные задания для проведения обобщающих занятий	18
Приложение 3. Учебные проекты (образцы программного кода)	21

1. Введение

Умение программировать необычайно востребовано в современном мире, не случайно подготовке специалистов в области программирования уделяется так много внимания в национальном проекте «Цифровая экономика». Кроме того, навыки программирования в настоящее время необходимы в самых различных специальностях, в том числе не связанных напрямую с IT-сферой.

Данная методическая разработка посвящена формированию у обучающихся интереса к информационным технологиям через изучение программирования, направлена на реализацию ФГОС и развитие интереса к техническим специальностям, выявление приоритетных интересов школьников и развитие их собственной одаренности. Реализация этого направления представлена в формате дополнительной образовательной программы «Программирование на Python: первые шаги».

Целевая аудитория разработки - обучающиеся 12-13 лет (7 класс), проявляющие интерес к информатике, мотивированные к углублению знаний по предмету, расширению кругозора, проецированию предметных знаний в прикладное направление.

Актуальность разработки обусловлена необходимостью формирования у детей алгоритмического мышления и навыков программирования, заинтересованности к техническим специальностям. Многие обучающиеся указанного возраста испытывают потребность в пропедевтическом изучении основ программирования, получении практических навыков создания приложений, с том числе с оконным интерфейсом.

Программа представляет последовательное и подробное описание этапов работы педагога по обучению детей основам программирования на языке Python, создания с его помощью консольных и оконных приложений игрового характера, являющихся значимыми и интересными для учащихся. У Python очень простой и понятный синтаксис, именно поэтому он наилучшим образом подходит для обучения детей данного возраста. Вместе с тем, это вполне профессиональный язык программирования, который широко используют в своей деятельности гиганты IT-индустрии. Благодаря постепенному нарастанию сложности занятий, ученики быстро осваивают интерфейс редактора и команды языка, получают представление об особенностях работы событийных программ и могут применить полученные знания при разработке собственных проектов. В процессе создания проектов обучающиеся учатся записывать алгоритм с помощью команд языка программирования, знакомятся с объектно-событийной моделью работы оконных приложений, что способствует развитию алгоритмического мышления, воображения, креативности, навыков поиска и устранения ошибок, реализовывает и развивает интерес к техническому творчеству.

2. Описание дополнительной образовательной программы «Программирование на Python: первые шаги»

Цель методической разработки: формирование и развитие у обучающихся интеллектуальных и практических компетенций в области программирования.

Задачи:

- сформировать у обучающихся представление об основных алгоритмических конструкциях и базовых структурах данных;
- познакомить с основными приемами работы в редакторе среды программирования;
- способствовать освоению базовых принципов процедурного программирования;
- познакомить с особенностями событийно-ориентированных программ;
- приобщать к практическому освоению создания программных продуктов;
- развивать алгоритмическое и креативное мышление;
- содействовать раскрытию творческого потенциала ребенка;
- способствовать воспитанию потребности в творческом труде, трудолюбия как высокой ценности в жизни;
- формировать позитивное отношение учащегося к собственному интеллектуальному развитию и воспитанию гражданской культуры личности;
- развивать умение работать в коллективе.

Новизна данной методической разработки заключается в том, что в программе «Программирование на Python: первые шаги» первоначальное знакомство обучающихся с основами программирования производится в аспекте их применения для создания практически значимых для обучающихся консольных и оконных приложений игрового характера. Дети получают возможность выразить себя через продукт, который они могут показать другим, так как объем и сложность проекта, выполнением которого завершается изучение каждого модуля, варьируется с учетом индивидуальных интересов и способностей обучающихся.

Кроме того, используемые в процессе обучения задачи в основном имеют ситуативный характер, формулируются не в виде стандартных упражнений на использование той или иной алгоритмической конструкции или типа данных, а в виде жизненных ситуаций, что отвечает компетентностному подходу и способствует формированию критического мышления и функциональной грамотности.

Обучающиеся, освоившие программу, овладеют основными приемами работы в среде программирования; получают представление о принципах создания программ с использованием основных алгоритмических конструкций, функций, строк, списков; получают опыт практического создания программ на языке Python; сформируют и разовьют коммуникативные навыки, необходимые для сотрудничества; разовьют алгоритмическое мышление и творческий

потенциал; смогут выполнить творческий мини-проект по программированию; сформируют представление о работе программиста.

В результате реализации данной разработки ожидается, что у обучающихся будут сформированы следующие личностные, метапредметные и предметные результаты.

Личностные результаты:

- развитие познавательных интересов и творческих способностей, креативного и алгоритмического мышления;
- владение первичными навыками анализа и критичной оценки получаемой информации;
- формирование адекватной самооценки и самопринятия;
- формирование способности увязать учебное содержание с собственным жизненным опытом, понять значимость подготовки в области информатики в условиях развития информационного общества;
- формирование готовности к повышению своего образовательного уровня и продолжению обучения с использованием средств и методов информатики и информационных технологий;
- формирование способности к волевому усилию и преодолению препятствий;
- умение продуктивно общаться и взаимодействовать в процессе совместной деятельности, учитывать позиции других участников деятельности;
- эстетическое отношение к миру, включая эстетику научного и технического творчества.

Метапредметные результаты:

- формирование умений самостоятельно планировать пути достижения целей; соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности, определять способы действий в рамках предложенных условий, корректировать свои действия в соответствии с изменяющейся ситуацией; оценивать правильность выполнения учебной задачи;
- владение основами самоконтроля, самооценки, принятия решений и осуществления осознанного выбора в учебной и познавательной деятельности;
- владение информационно-логическими умениями: устанавливать аналогии, причинно-следственные связи, строить логическое рассуждение, умозаключение и делать выводы;
- владение навыками познавательной, учебно-исследовательской и проектной деятельности, навыками разрешения проблем;
- владение основными универсальными умениями информационного характера: постановка и формулирование проблемы, поиск и выделение необходимой информации, самостоятельное создание алгоритмов деятельности при решении проблем творческого и поискового характера.

Предметные результаты:

- знание базовых принципов использования основных алгоритмических конструкций;

- владение основными инструментами и приемами работы в среде программирования;
- владение умением использовать различные типы данных: числа, строки, списки;
- формирование представлений об использовании функций для структурирования программного кода;
- формирование представлений об особенностях событийно-ориентированных программ;
- формирование базовых навыков и умений по соблюдению требований техники безопасности, гигиены и ресурсосбережения при работе со средствами информатизации.

В основу данной разработки положены следующие источники:

1. Веденева, О. А. Педагогические технологии в современном образовательном процессе: учебное пособие / Веденева О. А., Савва Л. И., Сайгушев Н. Я. - Москва: Мир науки, 2016.
2. Д. Бриггс. Python для детей. Самоучитель по программированию. – М.: Манн, Иванов и Фербер, 2017.
3. У. Сэнд, К. Сэнд. Hello, World! Занимательное программирование. – СПб.: Питер, 2016.

В программе реализуется практико-ориентированный, деятельностный подход, поэтому большинство заданий – практико-ориентированного характера, используется работа в парах, выполнение индивидуальных и парных проектов, консультации, обсуждения, самостоятельная практическая работа на занятиях.

Диагностика проводится в начале изучения курса (в форме беседы), в середине и в конце курса (выполнение ситуативных заданий), а также в форме самопроверки усвоения материала каждой темы (по результатам выполнения тестовых заданий), оценки и самооценки творческих проектов.

В разработке учитываются возрастные и психологические особенности школьников, предложенная программа лично ориентирована и допускает вариативность как в отношении распределения учебного времени, так и в выборе объема и сложности проектов, за счет чего педагог может адаптировать ее к конкретным условиям, учитывая индивидуальные особенности, развитие, интересы обучающихся.

Программа рассчитана на 32 часа и предлагает реализацию следующего содержания:

Тема 1: «Переменные, ввод и вывод данных» (4ч.): Язык Python. Среда IDLE. Первая программа. Вычисления и переменные. Ввод данных. Разница между числами и строками. Проект №1. Игра «Чепуха»

Тема 2: «Ветвления» (4ч.): Условная конструкция. Случайные числа. Графический интерфейс пользователя. Проект №2. Игра «Камень, ножницы, бумага»

Тема 3: «Циклы» (4ч.): Счетный цикл. Цикл с условием. Вмешательство в работу цикла. Проект №3. Игра «Пещера драконов»

Обобщающие занятия по темам 1-3 (2ч): работа с комплексными ситуативными заданиями (*Приложение 2*).

Тема 4: «Строки и списки» (4ч.): Списки. Строки и списки: сходство и различие. Обработка списков и строк. Проект №4. Игра «Гороскоп»

Тема 5: «Функции» (4ч.): Повторное использование кода с помощью функций. Параметры функции. Рисование в окне. Обработка событий. Проект №5. «Оживший смайлик»

Тема 6: «Итоговый проект» (4ч.): Проект №6. Игра «Угадай слово» (комплексный проект, создаваемый в рамках деловой игры «Студия разработчиков»)

Обобщающие занятия (2ч): работа с комплексными ситуативными заданиями (*Приложение 2*).

Творческий проект (4 ч.): Работа над творческим проектом. Презентация творческих проектов.

Элементы содержания, формы активности и контроля по каждому модулю подробно описаны в предлагаемой программе (*Приложение 1*).

Реализация учебных проектов по каждому модулю (листинг программного кода) приведена в *Приложении 3*.

3. Заключение

Методическая разработка «Программирование на Python: первые шаги» полностью соответствует реализации требований государства по развитию России. Стране нужны специалисты IT-сферы, умеющие использовать современные технологии программирования. Образовательная программа способствует формированию у детей заинтересованности к техническим специальностям, знакомит их с одним из наиболее востребованных языков программирования, основными алгоритмическими конструкциями, принципами создания событийно-ориентированных приложений.

Методическая разработка прошла успешную апробацию в рамках субботней школы для обучающихся 7 класса в СОГБОУИ «Лицей имени Кирилла и Мефодия», где используется начиная с 2018 года по настоящее время. В процессе апробации были сбалансированы по уровню сложности комплексы ситуативных задач и уточнено содержание проектных заданий по каждому модулю. По итогам реализации программы администрацией лицея получены положительные отзывы от обучающихся и их родителей.

Методическая разработка может быть полезной школьным учителям информатики, руководителям кружков технического направления в общеобразовательных учреждениях и учреждениях дополнительного образования, а также педагогам региональных центров работы с мотивированными и одаренными детьми для организации образовательных и проектных смен.

Приложения

Приложение 1. Дополнительная образовательная программа «Программирование на Python: первые шаги»

СМОЛЕНСКОЕ ОБЛАСТНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ С ИНТЕРНАТОМ
«ЛИЦЕЙ ИМЕНИ КИРИЛЛА И МЕФОДИЯ»

Дополнительная образовательная программа

Программирование на Python: первые шаги

Возрастная категория: 12-13 лет

Автор-составитель:
Сенчилова Ольга Константиновна,
учитель информатики.

2020 – 2021 уч. год

Содержание

Содержание.....	2
1. Пояснительная записка.....	2
1.1. Описание программы.....	2
1.2. Аннотация к программе.....	2
1.3. Цель и задачи программы.....	3
1.4. Планируемые результаты освоения программы.....	4
1.5. Основные методы и формы реализации содержания программы	5
1.6. Задания продуктивного и репродуктивного характера, выполняемые в рамках программы.....	6
1.7. Оценка результатов освоения программы.....	6
1.8. Требования к условиям организации образовательного процесса	6
2. Содержательная характеристика программы.....	6
2.1. Основные позиции	6
2.2. Содержательная характеристика	8
Список источников	10

1. Пояснительная записка

1.1. Описание программы

Направление: дополнительная образовательная программа технической направленности.

Целевая аудитория программы: программа рассчитана на обучающихся 12-13 лет, проявляющих интерес к информатике, мотивированных к углублению знаний по предмету, расширению кругозора, проецированию предметных знаний в прикладное направление.

Сроки реализации программы: программа рассчитана на 32 часа.

Ключевые понятия: среда программирования; переменные; типы данных: числа, строки, списки; условная конструкция; цикл; функция; случайные числа; графический интерфейс; обработка событий.

1.2. Аннотация к программе

Программа направлена на реализацию ФГОС, на формирование предметных умений через деятельность с опорой на личный опыт обучающихся, выявление приоритетных интересов школьников и развитие их собственной одаренности, формирование интереса к техническим специальностям, развитие креативности, логического мышления, а также навыков поиска и устранения ошибок.

Умение программировать необычайно востребовано в современном мире, не случайно подготовке специалистов в области программирования уделяется так много внимания в национальном проекте «Цифровая экономика». Кроме того, навыки программирования в настоящее время необходимы в самых различных специальностях, в том числе не связанных напрямую с IT-сферой.

Актуальность программы обусловлена необходимостью формирования у детей алгоритмического мышления и навыков программирования, заинтересованности к техническим специальностям. В связи с этим еще до начала изучения курса информатики основной школы им предоставляется возможность пропедевтического изучения основ программирования и получения практических навыков создания приложений, с том числе с оконным интерфейсом.

Новизна программы «Программирование на Python: первые шаги» определяется тем, что она предполагает первоначальное знакомство обучающихся с основами программирования в аспекте возможности их применения для создания практически значимых для обучающихся консольных и оконных приложений игрового характера.

Педагогическая целесообразность заключается в выявлении интереса обучающихся к знаниям и формировании устойчивого интереса к программированию с использованием языка Python. У Python очень простой и понятный синтаксис, именно поэтому он наилучшим образом подходит для обучения детей данного возраста. Вместе с тем, это вполне профессиональный язык программирования, который широко используют в своей деятельности гиганты IT-индустрии. Благодаря постепенному нарастанию сложности занятий, ученики быстро освоят интерфейс редактора и команды языка, получат представление об особенностях работы событийных программ и смогут применить полученные знания при разработке собственных проектов. В процессе создания проектов обучающиеся научатся записывать алгоритм с помощью команд языка программирования, ознакомятся с объектно-событийной моделью работы оконных приложений, что будет способствовать развитию алгоритмического мышления, воображения, креативности, реализовывать и развивать интерес к техническому творчеству.

Практическая значимость состоит в том, что обучающиеся получают навыки создания консольных и оконных приложений, которые позволят им разрабатывать собственные программы, в том числе в рамках выполнения проектов по робототехнике.

Отличительные особенности: программа использует компетентностный подход, большинство заданий имеют сюжетный характер, формулируются не в виде стандартных упражнений на использование той или иной алгоритмической конструкции или типа данных, а в виде житейских ситуаций. Помимо этого, дети получают возможность выразить себя через продукт, который они могут показать другим, так как изучение каждого модуля завершается выполнением проекта, объем и сложность которого варьируется с учетом индивидуальных интересов и способностей обучающихся.

1.3. Цель и задачи программы:

Цель: формирование и развитие у обучающихся интеллектуальных и практических компетенций в области программирования.

Задачи:

- сформировать у обучающихся представление об основных алгоритмических конструкциях и базовых структурах данных;
- познакомить с основными приемами работы в редакторе среды программирования;
- способствовать освоению базовых принципов процедурного программирования;
- познакомить с особенностями событийно-ориентированных программ;
- приобщать к практическому освоению создания программных продуктов;
- развивать алгоритмическое и креативное мышление;
- содействовать раскрытию творческого потенциала ребенка;
- способствовать воспитанию потребности в творческом труде, трудолюбия как высокой ценности в жизни;
- формировать позитивное отношение учащегося к собственному интеллектуальному развитию и воспитанию гражданской культуры личности;
- развивать умение работать в коллективе.

1.4 Планируемые результаты освоения программы

Обучающиеся, освоившие программу

- овладеют основными приемами работы в среде программирования;
- получают представление о принципах создания программ с использованием основных алгоритмических конструкций, функций, строк, списков;
- получают опыт практического создания программ на языке Python;
- сформируют и разовьют коммуникативные навыки, необходимые для сотрудничества;
- разовьют алгоритмическое мышление и творческий потенциал;
- смогут выполнить творческий мини-проект по программированию;
- сформируют представление о работе программиста.

В результате освоения данной программы ожидается, что у обучающихся будут сформированы следующие личностные, метапредметные и предметные результаты.

Личностные результаты:

- развитие познавательных интересов и творческих способностей, креативного и алгоритмического мышления;
- владение первичными навыками анализа и критичной оценки получаемой информации;
- формирование адекватной самооценки и самопринятия;
- формирование способности увязать учебное содержание с собственным жизненным опытом, понять значимость подготовки в области информатики в условиях развития информационного общества;
- формирование готовности к повышению своего образовательного уровня и продолжению обучения с использованием средств и методов информатики и информационных технологий;
- формирование способности к волевому усилию и преодолению препятствий;

- умение продуктивно общаться и взаимодействовать в процессе совместной деятельности, учитывать позиции других участников деятельности;
- эстетическое отношение к миру, включая эстетику научного и технического творчества.

Метапредметные результаты:

- формирование умений самостоятельно планировать пути достижения целей; соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности, определять способы действий в рамках предложенных условий, корректировать свои действия в соответствии с изменяющейся ситуацией; оценивать правильность выполнения учебной задачи;
- владение основами самоконтроля, самооценки, принятия решений и осуществления осознанного выбора в учебной и познавательной деятельности;
- владение информационно-логическими умениями: устанавливать аналогии, причинно-следственные связи, строить логическое рассуждение, умозаключение и делать выводы;
- владение навыками познавательной, учебно-исследовательской и проектной деятельности, навыками разрешения проблем;
- владение основными универсальными умениями информационного характера: постановка и формулирование проблемы, поиск и выделение необходимой информации, самостоятельное создание алгоритмов деятельности при решении проблем творческого и поискового характера.

Предметные результаты:

- знание базовых принципов использования основных алгоритмических конструкций;
- владение основными инструментами и приемами работы в среде программирования;
- владение умением использовать различные типы данных: числа, строки, списки;
- формирование представлений об использовании функций для структурирования программного кода;
- формирование представлений об особенностях событийно-ориентированных программ;
- формирование базовых навыков и умений по соблюдению требований техники безопасности, гигиены и ресурсосбережения при работе со средствами информатизации.

1.5 Основные методы и формы реализации содержания программы:

- инструктажи, беседы, разъяснения;
- теоретические лекции и "активное слушание";
- наглядные (визуализация алгоритмов);
- практические работы;
- аналитическая и исследовательская деятельность (информационный поиск, анализ данных, экспериментирование);
- проектная деятельность (моделирование и прототипирование);
- создание ситуации новизны, ситуации гарантированного успеха;

- индивидуальная работа и работа в парах.

Формы организации работы

Основной формой занятия является учебно-практическая деятельность. В программе реализуется практико-ориентированный, деятельностный подход, поэтому большинство заданий – практико-ориентированного характера, используется работа в парах, выполнение индивидуальных и парных проектов, консультации, обсуждения, самостоятельная практическая работа на занятиях.

Диагностика проводится в начале изучения курса (в форме беседы), в середине и в конце курса (выполнение ситуативных заданий), а также в форме самопроверки усвоения материала каждой темы (по результатам выполнения тестовых заданий), оценки и самооценки творческих проектов.

Во время выполнения работ необходимо уделять внимание правилам безопасного труда, организации рабочего места и санитарно-гигиеническим требованиям.

1.6 Задания продуктивного и репродуктивного характера, выполняемые в рамках программы

В ходе реализации программы каждый обучающийся выполнит следующие виды работ:

- *выполнение репродуктивных заданий* тестового и практического характера, позволяющих оценить степень усвоения материала;
- *информационный поиск*, предполагающий отбор и структурирование;
- *работа с ситуативными заданиями*, предполагающими анализ данных и рефлексия;
- *индивидуальные (парные) проекты*, предполагающие создание законченного продукта, содержательного и интересного для учащихся.

1.7. Оценка результатов освоения программы

В результате освоения программы обучающиеся получают следующие оценки результатов: оценка качества выполнения практических заданий по программе, в том числе комплексных ситуативных заданий; оценка качества выполнения творческого мини-проекта.

1.8 Требования к условиям организации образовательного процесса

Задействованность аудиторного фонда и оборудования: компьютерный класс (оборудование: компьютеры/ноутбуки, проектор и экран), время использования 32 часа.

2. Содержательная характеристика программы

2.1. Основные позиции

Содержательная часть программы включает в себя изучение основ программирования на языке Python (понятие переменной; типы данных число, строка, список; алгоритмические конструкции «следование», «ветвление» и «цикл»; оформление частей программы в виде функций; первичное представление о пользовательском интерфейсе и обработке событий). Часть тем

(типы данных, основные алгоритмические конструкции) являются пропедевтическими для базового курса информатики, создают необходимую базу для участия в предметной олимпиаде по информатике.

В программу включены задания репродуктивного и продуктивного характера, направленные на выявление приоритетных интересов школьников и развитие их способностей к программированию.

Программа рассчитана на 32 часа.

Содержание программы

Тема 1: «Переменные, ввод и вывод данных» (4ч.)

1. Язык Python. Среда IDLE. – 1ч
2. Первая программа. Вычисления и переменные – 1ч
3. Ввод данных. Разница между числами и строками – 1ч
4. Проект №1. Игра «Чепуха» – 1ч

Тема 2: «Ветвления» (4ч.)

1. Условная конструкция – 2ч
2. Случайные числа. Графический интерфейс пользователя – 1ч
3. Проект №2. Игра «Камень, ножницы, бумага» – 1ч

Тема 3: «Циклы» (4ч.)

1. Счетный цикл. Цикл с условием. Вмешательство в работу цикла – 2ч
2. Проект №3. Игра «Пещера драконов» – 2ч

Обобщающие занятия по темам 1-3 (2ч)

Работа с комплексными ситуативными заданиями.

Тема 4: «Строки и списки» (4ч.)

1. Списки. Строки и списки: сходство и различие. Обработка списков и строк – 2ч
2. Проект №4. Игра «Гороскоп» - 2ч

Тема 5: «Функции» (4ч.)

1. Повторное использование кода с помощью функций. Параметры функции – 1ч
2. Рисование в окне. Обработка событий – 1ч
3. Проект №5. «Оживший смайлик»– 2ч

Тема 6: «Итоговый проект» (4ч.)

1. Проект №6. Игра «Угадай слово» – 4ч

Обобщающие занятия (2ч)

Работа с комплексными ситуативными заданиями.

Творческий проект (4 ч.)

1. Работа над творческим проектом – 2 ч.
2. Подготовка к презентации проектов – 1 ч.
3. Презентация творческих проектов – 1 ч.

2.2. Содержательная характеристика

№ п/п	Тема	Содержание	Кол-во часов	Формы активности/ контроля	Ресурсы
1	Язык Python. Среда IDLE	Анализ потребностей в обучении. Программирование – необходимый навык для жизни в современном мире. Язык Python. Среда программирования IDLE: преимущества для начинающих. Установка и настройка Python и IDLE.	1	Беседа. Фронтальная практическая работа «Установка среды программирования»	Видеофрагмент «Кто такой программист»
2	Первая программа. Вычисления и переменные	Интерфейс IDLE. Сохранение и запуск программ. Оператор вывода. Вычисления в консоли. Вычисления с числовыми переменными.	1	Практические задания «Вычисления в Python»	Презентация «Установка Python. Работа в IDLE»
3	Ввод данных. Разница между числами и строками	Числовые и строковый типы данных. Ввод и вывод строк. Умножение строки на число. Преобразование типов. Ввод целых и вещественных чисел.	1	Демонстрация. Практические задания	Раздаточный материал «Вычислительные задачи»
4	Проект №1. Игра «Чепуха»	Разработка линейной программы, которая запрашивает данные строкового типа и выводит текст с использованием данных слов.	1	Оценивание первичного понимания материала. Работа над проектом	Тест №1 «Переменные, ввод и вывод данных»
5	Условная конструкция	Конструкция if. Блоки. Условия и сравнение значений. Конструкция if...else. Выражение elif. Сложные условия	2	Лекция. Практические задания. Оценивание первичного понимания материала	Презентация «Ветвление». Тест №2 «Ветвления»
6	Случайные числа. Графический интерфейс пользователя	Библиотеки. Библиотека random и функция randint() Библиотека tkinter: окна, кнопки, вставка изображений	1	Фронтальная лабораторная работа	Индивидуальный раздаточный материал
7	Проект №2. Игра «Камень, ножницы, бумага»	Моделирование игры с соперником при помощи случайных чисел. Использование условной конструкции для проверки выигрыша и проигрыша. Создание оконного проекта с использованием полей ввода и кнопок.	1	Групповая работа по составлению алгоритма. Работа над проектом	Видеоролик (планируемый результат). Сценарий работы программы

8	Счетный цикл. Цикл с условием. Вмешательство в работу цикла	Цикл for. Функция range(), ее параметры. Цикл while. Инструкции continue и break	2	Лекция. Практические задания. Оценивание первичного понимания	Индивидуальный раздаточный материал. Тест №3 «Циклы»
9	Проект №3. Игра «Пещера драконов»	Используется цикл для повторения игровых действий: а) указанное количество раз (выводится счет); б) до достижения определенного результата (или поражения).	2	Эвристическая беседа. Групповая работа над проектом (по выбранному типу цикла)	Видеоролик (планируемый результат). Варианты сценария работы программы
10	Работа с комплексными ситуативными заданиями	Сюжетные задания на ввод и вывод данных, составление программ с использованием ветвлений и циклов. Анализ результатов работы	2	Практическая работа. Рефлексия	Раздаточный материал (ситуативные задачи)
11	Списки. Строки и списки: сходство и различие. Обработка списков и строк	Список. Создание списка. Строки и списки: сходство и различие. Доступ к элементам, перебор элементов в цикле. Функции обработки строк и списков. Срезы	2	Лекция Фронтальная лаб. работа Практические задания Оценивание первичного понимания	Презентация Индивидуальный раздаточный материал Тест №4 «Строки и списки»
12	Проект №4. Игра «Гороскоп»	Использование списков для хранения данных, случайный выбор данных из списка. Комбинирование данных для вывода в строках.	2	Эвристическая беседа. Работа над проектом	Видеоролик (планируемый результат). Сценарий работы программы
13	Повторное использование кода с помощью функций. Параметры функции	Применение функций. Строение функции. Вызов функции. Передача параметров. Функции с несколькими аргументами	1	Лекция. Фронтальная лаб. работа. Оценивание первичного понимания	Презентация Тест №5 «Функции»
14	Рисование в окне. Обработка событий	Библиотека tkinter: рисование на холсте, создание функций обработки событий	1	Фронтальная лабораторная работа	Раздаточный материал «Команды графики»
15	Проект №5. «Оживший смайлик»	Создание изображения из графических примитивов. Оформление в виде функций кода, рисующего отдельные части изображения. Программирование реакции на события (нажатие	2	Работа в группах: создание функций и объединение их в общий проект. Практическая	Раздаточный материал «Смайлик в системе координат» Проектные задания для

		клавиш)		работа: до- бавление ин- терактивности	<i>творческого проекта</i>
16	Проект №6. Игра «Угадай слово»	Комплексный проект, ори- ентированный на получение законченного программного продукта, содержательного и интересного для обучаю- щихся.	4	Деловая игра «Студия разра- ботчиков»	Проектное задание. Раздаточный материал для групп.
17	Работа с ком- плексными ситуа- тивными задани- ями	Сюжетные задания на со- ставление программ с ис- пользованием ветвлений, циклов, функций, обраба- тывающих числа, строки, списки. Анализ результатов работы	2	Практическая работа. Рефлексия	Раздаточный материал (ситуатив- ные задачи)
18	Работа над твор- ческим проектом	Постановка задачи. Со- ставление плана работы. Поиск и анализ информа- ции. Составление алгоритма и его реализация.	2	Работа с ис- точниками информации. Работа по ин- дивидуально- му плану	Проектные задания. Примеры: «Текстовый квест», «Ин- терактивная анимация»
19	Подготовка к пре- зентации проектов	Размещение проектов в об- щей папке в облачном хра- нилище. Создание аннота- ций к проектам и размеще- ние их в общей презента- ции.	1	Работа по ин- дивидуально- му плану	Папка в об- лачном хра- нилище. Общая пре- зентация.
20	Презентация творческих проек- тов	Презентация проектов, вы- полненных по индивиду- альным проектным задани- ям	1	Презентация проектов. Са- мооценива- ние. Взаимо- оценивание.	Листы для оценивания и самооце- нивания

Список источников

4. Д. Бриггс. Python для детей. Самоучитель по программированию. – М.: Манн, Иванов и Фербер, 2017.
5. У. Сэнд, К. Сэнд. Hello, World! Занимательное программирование. – СПб.: Питер, 2016.

Приложение 2. Комплексные ситуативные задания для проведения обобщающих занятий

Работа №1 (Обобщающее занятие по темам 1-3)

В детском лагере была сформирована команда юных программистов. Ребята автоматизировали работу лагеря. Тебя позвали в команду программистов. Ознакомьтесь с техническим заданием и напишите программы. *Замечание: Указанные в примерах числа даны только для образца. Настоящая программа должна работать и с другими числами!*

Техническое задание

Задание 1. Заезд.

Напишите программу, которая считает число свободных мест в лагере. В лагере имеется 10 комнат, в каждой по 6 мест. Программа получает на вход число купленных путевок.

Пример работы программы:

Сколько путевок куплено?

34

Осталось 26 мест

Задание 2. Время обеда

В лагере небольшая столовая, поэтому дети ходят на обед в разное время. В 12 часов приходят дети, которые живут в комнатах с четными номерами, а в 13 часов – с нечетными. Напишите справочную программу, которая по номеру комнаты сообщает ребенку, во сколько ему идти на обед.

Пример работы программы:

Твой номер комнаты:

7

Обед в 13 часов

Задание 3. Найди свой отряд

Напишите программу, которая выдает номер отряда, в котором будет состоять новый участник смены. Программа получает на вход возраст ребенка. Его распределяют в 1 отряд, если он младше 10 лет, во 2 отряд, если ему от 10 до 12 лет, и в 3 отряд, если он старше 12 лет.

Пример работы программы:

Сколько лет ребенку?

12

2 отряд

Задание 4. Отчет директора

Вожатые ежедневно составляют отчет о количестве проведенных мероприятий за день. Смена длится 21 день. В конце смены директор лагеря вводит в компьютер каждое из этих чисел и получает общее количество мероприятий, проведенных за смену. К сожалению, компьютер оказался заражен вирусом, который стер часть написанного кода. Восстановите код программы.

```
s =
for i
    print('Введите количество мероприятий в', i, 'день:')
    ...
    ...
print('Всего проведено', s, 'мероприятий')
```

Работа №2 (Обобщающее занятие по темам 1-6)

Вы – участник команды школьников, решивших написать приложение «Скакалочка» для организации тренировок в прыжках со скакалкой. Вам поручено разработать несколько модулей, которые будут использованы в коде будущего приложения. Напишите программу для решения каждой из предложенных задач. *Замечание: Указанные в примерах данные приведены для образца. Настоящая программа должна работать и с другими данными!*

Задание 1. Регистрация

При первом входе в приложение пользователь должен ввести свое имя. Программа должна запомнить это имя и вывести приветственную фразу вида «Хороших тренировок, (Имя)!».

Пример работы программы:

Введите имя:

Мария

Хороших тренировок, Мария!

Задание 2. Вызов

В приложении будет возможность получить задание на день: количество прыжков, которое будет выбираться случайным образом в зависимости от пола: от 300 до 450 включительно для мальчиков и от 350 до 500 для девочек. Пока этот модуль проходит тестирование, поэтому ввод пола пользователя выполняется с клавиатуры. Напишите программу, которая получает на вход одно из двух значений: “м” или “ж”, и выдает количество прыжков.

Пример работы программы:

Твой пол:

ж

Прыжков на сегодня: 463

Задание 3. Статистика

Приложение хранит результаты тренировок в списке `results`. Считая, что список `results` уже существует и заполнен какими-то значениями, напишите команды, в результате выполнения которых на экран будет выведено:

- 1) количество проведенных тренировок;
- 2) суммарное количество прыжков за все тренировки;
- 3) среднее количество прыжков за одну тренировку;
- 4) максимальное количество прыжков, сделанных за время тренировок;
- 5) сколько прыжков сделано во время последней тренировки.

Считаем, что пока этот модуль проходит тестирование. Поэтому вы можете задать список `results` в начале программы и заполнить его произвольными числами. Например:

```
results = [220, 300, 410, 420, 280, 320, 340, 190, 340, 400]
```

Задание 4. Тренировка

Результат тренировки, вносимый в список `results`, определяется следующим образом. Пользователь сообщает программе количество выполненных за тренировку подходов, а затем вносит в программу количество прыжков, сделанных во время каждого из них. Программа должна вычислить суммарное количество прыжков, сделанных за всю тренировку. Этот результат нужно вывести на экран, а также добавить полученное число к списку `results`. Считаем, что этот модуль проходит тестирование, поэтому пока создайте пустой либо заполненный числами список `results` в начале программы. Например: `results = []`

Пример работы программы:

Количество подходов:

3

Сделано прыжков:

110

74

92

Прыжков за тренировку: 276

Задание 5. Результативность

После тренировки приложение выводит комментарий о ее результативности. В переменной S хранится суммарное количество выполненных за тренировку прыжков (считаем, что пока этот модуль проходит тестирование, поэтому выполните ввод S с клавиатуры). Напишите программу, которая в зависимости от значения S выводит одно из сообщений:

- «Отлично», если S больше 400;
- «Хорошо», если S от 200 до 400 включительно;
- «Слабовато», если S меньше 200.

Оформите этот фрагмент кода в виде функции, которая принимает значение S в качестве аргумента.

Задание 6. Награды

За некоторые действия пользователь будет получать в приложении награды в свою коллекцию. Например, если суммарное количество выполненных за тренировку прыжков заканчивается на два нуля, выдается награда «Круглое число», а если оно является трехзначным числом, у которого равны первая и последняя цифры, то выдается награда «Симметрия».

Напишите программу, которая в зависимости от значения S (суммарного количества выполненных за тренировку прыжков) определяет, получит ли пользователь одну из этих наград, и выводит соответствующее сообщение. Если никакая награда пользователю не положена, то выводить ничего не нужно.

Считаем, что пока этот модуль проходит тестирование, поэтому ввод S выполните с клавиатуры.

Пример работы программы:

Сколько сделано прыжков?

282

Вы получили награду «Симметрия»!

Задание 7*. Почти приложение

На основе решений заданий 1-6 составьте единую программу, в которой вначале проводится регистрация пользователя с вводом имени и пола, а затем начинается бесконечный цикл (либо цикл, который после каждого выполнения спрашивает пользователя, пора ли завершать работу). В этом цикле программа: предлагает задание на день; спрашивает у пользователя результаты тренировки и запоминает их в списке `results`, выводит комментарий о результативности тренировки, если есть – то о полученных наградах; выводит статистику тренировок за все время работы.

Приложение 3. Учебные проекты (образцы программного кода)

Проект №1. Игра «Чепуха»

```
print("Поиграем в Чепуху?")
print("Отвечай на вопросы в нужном падеже")
print("Какой?")
a = input()
print("Какая?")
b = input()
print("Какого?")
c = input()
print("Сколько? Введи цифру:")
x = int(input())
print("Какой?")
d = input()
print("Какой?")
e = input()
print("Какая?")
f = input()
print("Жили-были", a, "дед да", b, "баба. И решили они испечь",
      c, "колобка. ")
print("Пекли", x, "часов. Получился колобок", d, "и", e + ".")
print("Дед с бабушкой порадоваться не могли,", f, "стала их
жизнь.")
```

Проект №2. Игра «Камень, ножницы, бумага»

Консольная версия

```
import random
print('Твой выбор: камень, ножницы или бумага?')
answer = input().lower()
# 1 - камень, 2 - ножницы, 3 - бумага, 0 - ошибка
if answer == 'камень':
    a = 1
elif answer == 'ножницы':
    a = 2
elif answer == 'бумага':
    a = 3
else:
    a = 0
b = random.randint(1, 3)
if b == 1:
    print('Компьютер выбрал камень')
elif b == 2:
    print('Компьютер выбрал ножницы')
else:
    print('Компьютер выбрал бумагу')
# r - фраза с результатом игры
r = 'Ошибка. Сделайте выбор заново'
if a == b:
    r = 'Ничья'
elif (a==1 and b==2) or (a==2 and b==3) or (a==3 and b==1):
```

```

    r = "Вы победили!"
elif (a==2 and b==1) or (a==3 and b==2) or (a==1 and b==3):
    r = "Победил компьютер"
print(r)

```

Оконное приложение

```

import random
from tkinter import *

def mycom():
    # считываем, что написал пользователь, переводя все буквы в
    # маленькие
    answer = edt.get().lower()
    # превращаем ответ в число: 1-камень, 2-ножницы, 3-бумага
    if answer == 'камень':
        a = 1
    elif answer == 'ножницы':
        a = 2
    elif answer == 'бумага':
        a = 3
    else:
        a = 0
    # выбор компьютера - случайное число, 1-камень, 2-ножницы,
    # 3-бумага
    b = random.randint(1,3)
    # изменяем текст в третьей метке - пишем, что выбрал компью-
    # тер
    if b == 1:
        lab3.config(text = 'Компьютер выбрал камень')
    elif b == 2:
        lab3.config(text = 'Компьютер выбрал ножницы')
    else:
        lab3.config(text = 'Компьютер выбрал бумагу')
    # В переменной r появится сообщение о результате игры или
    # ошибке
    r = 'Ошибка. Сделайте выбор заново'
    # проверяем, какой результат игры
    if (a == 1 and b == 1) or (a == 2 and b == 2) or (a == 3 and
b == 3):
        r = 'ничья'
    if (a == 1 and b == 2) or (a == 2 and b == 3) or (a == 3 and
b == 1):
        r = 'Поздравляю, ты победил!'
    if (a == 1 and b == 3) or (a == 2 and b == 1) or (a == 3 and
b == 2):
        r = 'Неудача, победил компьютер.'
    # изменяем текст во второй метке - пишем результат игры
    lab2.config(text = r)

window = Tk()
window.geometry('200x150')
#пишем в окне вопрос
lab1 = Label(window, text='Камень, ножницы или бумага?')

```

```

lab1.pack()
#создаем поле ввода - сюда игрок напишет, что он выбрал
edt = Entry(window, width = 20)
edt.pack()
#создаем кнопку для запуска игры
btn = Button(window, text='Проверить', command=mycom)
# нажатие на кнопку будет приводить к вызову кода, которому мы
дали имя mycom
# и записали в начале программы - такой код называется ФУНКЦИЕЙ
btn.pack()
#создаем пустую надпись номер 2 - здесь появится результат игры
lab2 = Label(window, text='')
lab2.pack()
#создаем пустую надпись номер 3 - здесь будет написано, что вы-
брал компьютер
lab3 = Label(window, text='')
lab3.pack()
window.mainloop() #главный цикл: создается окно, которое ждет
действий игрока

```

Проект №3. Игра «Пещера драконов»

```

import random

print('Игра "Пещера драконов"')
hp = 100
s = 0
print('Вы в пещере')
while hp > 0:
    print('Здоровье:', hp, 'Деньги:', s)
    print('Из темноты выходит дракон...')
    print('Выбирай: 1 - атаковать, 0 - убежать')
    answer = input()
    if answer == '0':
        break
    else:
        result = random.randint(0,1)
        if result == 1:
            m = random.randint(25, 100)
            print('Победа! Найдено монет', m)
            s = s + m
        else:
            print('Ты ранен, а дракон унес сокровище в темно-
ту...')
            hp = hp - random.randint(10, 20)
if hp > 0:
    print('Ты сумел унести ноги и заработал', s, 'монет!')
else:
    print('Ты остался в пещере навсегда... и сокровища тоже')

```

Проект №4. Игра «Гороскоп»

Консольная версия

```
import random

data = ["сегодня вас ждет успех в делах",
        "держитесь подальше от шумных компаний",
        "не начинайте ничего нового сегодня",
        "случайное знакомство может обернуться крепкой дружбой"]
print("Введите имя")
name = input()
print("Ваш знак зодиака:")
znak = input()
n = len(data)
k = random.randint(0, n-1)
print(name + ", " + data[k])
```

Оконное приложение

```
import random
from tkinter import *

data = ["сегодня вас ждет успех в делах",
        "держитесь подальше от шумных компаний",
        "не начинайте ничего нового сегодня",
        "случайное знакомство может обернуться крепкой дружбой"]

def fate():
    name1 = name.get()
    n = len(data)
    k = random.randint(0, n-1)
    labelgoroskop.config(text=name1+', '+data[k])

root = Tk()
root.title("Гороскоп")
root.geometry("500x680")
labelname = Label(text='Введите имя: ')
labelname.pack()
name = Entry()
name.pack()
labelznak = Label(text='Ваш знак зодиака: ')
labelznak.pack()
znak = Entry()
znak.pack()
label_1=Label()
label_1.pack()
btn = Button(text="Узнать свою судьбу!", background= "#8470ff",
foreground= "#ffff00", height="0", width="25", command= fate)
btn.pack()
label_2=Label()
label_2.pack()
labelgoroskop=Label() #для предсказания
labelgoroskop.pack()
label_3=Label(justify=CENTER)
```

```

label_3.pack()
img1 = PhotoImage(file='goroskop.gif')
label_img = Label(image=img1)
label_img.pack(side=BOTTOM)
root.mainloop()

```

Проект №5. «Оживший смайлик»

```

import time
from tkinter import *

def eye_left():
    for x in range(10):
        canvas.move(l_eye, -1, 0)
        canvas.move(r_eye, -1, 0)
        window.update()
        time.sleep(0.05)

def eye_right():
    for x in range(10):
        canvas.move(l_eye, 1, 0)
        canvas.move(r_eye, 1, 0)
        window.update()
        time.sleep(0.05)

def smile():
    canvas.itemconfig(mouth, start=210, extent=120)
    window.update()

def dissmile():
    canvas.itemconfig(mouth, start=230, extent=80)
    window.update()

# создаем окно
window = Tk()
window.geometry("600x600")
# создаем холст для рисования
canvas = Canvas(window, width=600, height=600, bg="lightgreen")
# создаем объекты на холсте
canvas.create_oval(100,100,500,500,fill="yellow")
canvas.create_oval(200,200,260,280,fill="white")
canvas.create_oval(340,200,400,280,fill="white")
l_eye=canvas.create_oval(220,240,250,270,fill="blue",outline="blue")
r_eye=canvas.create_oval(360,240,390,270,fill="blue",outline="blue")
mouth=canvas.create_arc(200,200,400,400,outline="red",width=5,
                        start=230,extent=80,style=ARC)

# размещаем холст в окне
canvas.pack()
# движение
for x in range(10):

```

```

    eye_left()
    eye_right()
    smile()
    time.sleep(1)
    dissmile()
    time.sleep(1)

window.mainloop() # запуск основного цикла

```

Проект №6. Игра «Угадай слово»

```

import random
words = ["муравей", "бабуин", "барсук", "медведь", "бобр", "вер-
блюд"]

def getRandomWord():
    k = random.randint(0, len(words)-1)
    return words[k]

def printWord(good_letters, secret_word):
    s = "" # будем собирать слово в переменной s, пока там пусто
    for x in secret_word:
        if x in good_letters:
            s = s + " " + x
        else:
            s = s + " " + "_"

def displayBoard(bad_letters, good_letters, secret_word):
    print('Тема: "Животные"')
    print('Осталось попыток:', 10-len(bad_letters))
    print('Неправильные буквы:', bad_letters)
    print('Слово:')
    printWord(good_letters, secret_word) # функция печати слова

def getLetter(bad_letters, good_letters):
    good = 0 # пометка о правильности символа, 0 -нет, 1- да
    while good == 0:
        print("Введите букву")
        x = input()
        if x in good_letters or x in bad_letters:
            print("Эта буква уже называлась")
        elif x not in 'ёйцукенгшщзхъфывапролджэячсмитьбю': # ес-
ли x нет в такой строке (состоящей из всех маленьких русских
букв), значит, это неподходящий символ
            print('Введите маленькую русскую букву')
        elif len(x) != 1: # если введена не
буква, а несколько букв
            print('Можно вводить только одну букву')
        else:
            good = 1
    return x

```

```

secret_word = getRandomWord()
bad_letters = "          # строка для хранения неправильных букв
good_letters = "        # строка для хранения уже угаданных букв
game_over = 0          # пометка об окончании игры, 0 – нет, 1 - да
while game_over == 0:  # пока игра не окончена
    displayBoard(bad_letters, good_letters, secret_word) # функция показа "игрового поля"
    letter = getLetter(bad_letters, good_letters)      # спросить
    новую букву (с проверками!)
    if letter in secret_word:                          # если буква есть в слове
        good_letters = good_letters + letter # добавить ее к
        good_letters
        found_all = 1                                # пометка, все ли буквы угаданы, 0 -
        нет, 1 -да (предполагаем, что да)
        for x in secret_word:
            if x not in good_letters:
                found_all = 0                        # меняем значение на 0
        if found_all == 1:                            # если все буквы угаданы, то
            print("Ура! Вы угадали слово", secret_word)
            game_over = 1
    else:                                              # если буквы нет в слове
        bad_letters = bad_letters + letter
        if len(bad_letters) == 10:
            print("Увы! Вы не угадали слово", secret_word)
            game_over = 1
print("Игра окончена")

```