

ИНДИВИДУАЛЬНЫЙ ПРЕДПРИНИМАТЕЛЬ
ВИХАРЕВА МАРЬЯ АЛЕКСАНДРОВНА



ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА

«Курс программирования Python Start»

возраст обучающихся: 12 – 14 лет

срок реализации: 2 года (144 часа)

наполняемость группы: 6 – 12 человек

Автор-составитель: Вихарева М.А.

Мирный, 2025 год

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА.

Программа составлена на основе программы курса «Курс программирования Python Start» (К. Нагайцева, А. Спирин, Н. Основенко) Международной школы программирования и математики «Алгоритмика» в соответствии с федеральными нормативными правовыми актами в области дополнительного образования, государственными требованиями к образовательным программам системы дополнительного образования детей:

- Федеральный Закон от 29.12.2012 г. № 273-ФЗ «Об образовании в Российской Федерации»;
- Концепция развития дополнительного образования детей до 2030 года (распоряжение Правительства РФ от 31.03.2022 г. № 678-р);
- Приказ Министерства просвещения Российской Федерации от 27.07.2022 г. № 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;
- Приказ Министерства просвещения Российской Федерации от 03.09.2019 № 467 «Об утверждении целевой модели развития региональных систем дополнительного образования»;
- Приказ Минобрнауки России от 23.08.2017 № 816 «Об утверждении Порядка применения организациями, осуществляющими образовательную деятельность, электронного обучения, дистанционных образовательных технологий при реализации образовательных программ»;
- Методические рекомендации по проектированию дополнительных общеразвивающих программ (включая разноуровневые программы) (Приложение к письму Департамента государственной политики в сфере воспитания детей и молодежи Министерства образования и науки РФ от 18.11.2015 г. №09-3242);
- Методические рекомендации по реализации образовательных программ начального общего, основного общего, среднего общего образования, образовательных программ среднего профессионального образования и дополнительных общеобразовательных программ с применением электронного обучения и дистанционных образовательных технологий от 20.03.2020 г.;
- Постановление Главного государственного санитарного врача РФ от 28.01.2021 г. №2 «Об утверждении СанПиН 1.2.3.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания».

Направленность программы: техническая.

Уровень программы – стартовый.

Новизна программы заключается в превалировании прикладного аспекта над фундаментальным, что позволяет изучать программирование с нуля без значительной опоры на математику, демонстрировать применение полученных знаний в решении задач из реальной жизни и мотивировать учеников с разным уровнем подготовки.

Актуальность программы заключается в:

- формировании основных навыков программирования на языке Python;
- формировании и развитии навыков работы с информацией в условиях модернизации образования;

- формировании навыков предоставления и получения обратной связи;
- формировании понимания информационной безопасности;
- формировании навыков управления проектами;
- развитии цифровой и функциональной грамотности;
- развитии алгоритмического и креативного мышления, творчества через создание собственных проектов с использованием цифровых ресурсов;
- необходимости расширения кругозора школьника.

Педагогическая целесообразность: обучение осуществляется в рамках ролевой игры, сюжетная линия которой охватывает весь курс. Курс построен таким образом, что даёт возможность учащимся пройти через различные этапы работы в большой IT-компании. Ученик проходит путь от собеседования и начала карьеры стажёра-разработчика (Intern Python Developer) до ведущего разработчика (Senior Python Developer). Данные задачи не только способствуют формированию предметных компетенций, но ещё и формируют взгляд ученика на мир, раскрывают значение soft skills, учат грамотному использованию компьютера и готовят к жизни в цифровом обществе.

Использование компьютерных технологий в работе с детьми среднего школьного возраста является стремительно развивающейся методикой в образовании во всём мире. С её помощью можно более эффективно решать образовательные задачи, которые будут способствовать качественному улучшению обучения ребёнка в школе.

Цель программы – способствовать формированию алгоритмического и критического мышления, навыков проектной деятельности через использование языка программирования Python как инструмента для реализации идей в области своих интересов.

Задачи 1-ого года обучения:

Обучающие:

- познакомить с базовыми понятиями основ программирования;
- познакомить с особенностями синтаксиса языка программирования Python;
- познакомить с принципами объектно-ориентированного программирования;
- сформировать навык правильного оформления кода;
- сформировать навык использования итеративного подхода при решении различных задач;
- формировать навык решения задач и разработки проектов с помощью Python;
- формировать навык разработки игр с помощью языка программирования Python;
- формировать навык тестирования и оптимизации созданных проектов.

Развивающие:

- формировать и развивать логическое, алгоритмическое, критическое, креативное и пространственное мышление;
- формировать навык публичного выступления и презентации;
- формировать навык планирования деятельности и основ тайм-менеджмента;
- формировать и расширять словарный запас на английском языке;
- формировать функциональную грамотность;
- расширять кругозор, развивать память, внимание, творческое воображение, абстрактно-логические и наглядно-образные виды мышления и типы памяти, основные мыслительные операции и свойства внимания;

- совершенствовать диалогическую речь учащихся: уметь слушать собеседника, понимать вопросы, смысл знаний, уметь задавать вопросы, отвечать на них.

Воспитательные:

- воспитывать у учащихся потребность в сотрудничестве, взаимодействии со сверстниками, умение подчинять свои интересы определённым правилам, развивать эмоциональный интеллект;
- формировать умение давать качественную обратную связь и реагировать на неё;
- формировать информационную культуру;
- формировать умение самостоятельно находить и изучать информацию и доносить её до других.

Задачи 2-ого года обучения:

Обучающие:

- сформировать навык правильного оформления кода;
- формировать навык решения задач и разработки проектов с помощью Python;
- познакомить с профессиональными библиотеками языка Python (PyQt, PyGame, Pillow) и применить их на практике;
- сформировать навык использования итеративного подхода при решении различных задач;
- сформировать навык работы в профессиональном сообществе (на примере сообщества учеников «Алгоритмики» «Зал славы» и международного сервиса для публикации IT-проектов GitHub);
- познакомить с профессиональными инструментами разработчика (работа с текстовыми и графическими файлами, использование игровой библиотеки, GitHub);
- формировать навык разработки игр с помощью языка программирования Python;
- формировать навык тестирования и оптимизации созданных проектов.

Развивающие:

- формировать и развивать логическое, алгоритмическое, критическое, креативное и пространственное мышление;
- формировать навык публичного выступления и презентации;
- формировать навык планирования деятельности и основ тайм-менеджмента;
- Формировать и расширять словарный запас на английском языке;
- формировать функциональную грамотность;
- расширять кругозор, развивать память, внимание, творческое воображение, абстрактно-логические и наглядно-образные виды мышления и типы памяти, основные мыслительные операции и свойства внимания;
- совершенствовать диалогическую речь учащихся: уметь слушать собеседника, понимать вопросы, смысл знаний, уметь задавать вопросы, отвечать на них.

Воспитательные:

- воспитывать у учащихся потребность в сотрудничестве, взаимодействии со сверстниками, умение подчинять свои интересы определённым правилам, развивать эмоциональный интеллект;
- формировать умение давать качественную обратную связь и реагировать на неё;
- формировать информационную культуру;

- формировать умение самостоятельно находить и изучать информацию и доносить её до других.

Отличительная особенность программы: в рамках курса учащиеся освоят базовые принципы программирования на языке Python через создание собственных проектов и решение задач разной сложности (от исправления ошибки в чужой программе до запуска beta-версии своего продукта).

Особенность курса проявляется в том, что, изучив необходимый теоретический материал, учащимся предлагается самостоятельно решить практическую задачу, представленную в виде адаптированного кейса из реальной жизни. Для этого учащиеся проходят через следующие этапы: определяют и составляют последовательность действий, ведущую к цели, и записывают её с помощью формального языка. Полученные знания учащиеся применяют при создании собственных проектов, которые защищаются перед другими учениками, педагогами и родителями.

Адресат программы: обучение ведется в группах, которые комплектуются из обучающихся 12-14 лет. Отбор на курс предусмотрен только по возрасту.

Формы организации образовательного процесса:

Дети занимаются в кабинете с педагогом (групповая форма занятий), самостоятельно работая за компьютером (не более 20 минут за все занятие). Основные формы работы – индивидуальная, групповая и фронтальная.

Специфика форм обучения:

- обучение от общего к частному (дедуктивный метод) и от частного к общему (индуктивный);
- поощрение вопросов и свободных высказываний по теме;
- уважение и внимание к каждому ученику;
- создание мотивационной среды обучения;
- создание условий для дискуссий и развития мышления учеников при достижении учебных целей вместо простого одностороннего объяснения темы преподавателем.

Занятие состоит из следующих блоков:

- повторение и определение целей на урок (5 минут);
- изучение нового материала (20 минут);
- практика (15 минут);
- разминка (10 минут);
- изучение нового материала (10 минут);
- практика (20 минут);
- рефлексия: подведение итогов занятия (10 минут).

Объем и срок освоения программы: дополнительная общеобразовательная общеразвивающая программа «Курс программирования Python Start» рассчитана на 2 года (144 часа; 72 занятия).

Режим занятий: 1 занятие в неделю по 2 академических часа.

Ожидаемые результаты.

Достижение цели и задач образовательной программы предполагает получение следующих результатов:

1-ый год обучения

Планируемые результаты	Способ достижения	Критерий достижения образовательного результата
Предметные навыки		
Формирование знаний об основах программирования, в т. ч. алгоритмических конструкциях, логических значениях и операциях.	Изучение теоретического материала совместно с учителем, выполнение заданий на платформе с постепенным увеличением сложности.	Ученик оперирует базовыми понятиями основ программирования: алгоритм, цикл, условие, функция, переменная, структуры данных и т. д.
Знакомство с синтаксисом языка программирования Python.	Изучение теоретического материала совместно с учителем, выполнение заданий на платформе с постепенным увеличением сложности.	Ученик самостоятельно пишет программы, используя язык программирования Python.
Развитие умений применять Python для решения задач практического характера и задач из смежных дисциплин.	Выполнение заданий на платформе с постепенным увеличением сложности.	Ученик самостоятельно пишет программы на языке Python, решающие задачи практического характера.
Следование стандарту при оформлении кода.	Изучение теоретического материала совместно с учителем, выполнение заданий на платформе.	Ученик оформляет код в соответствии с общепринятыми стандартами и использует комментарии.
Умение тестировать программу и находить в ней ошибки.	Выполнение заданий на платформе с постепенным увеличением сложности.	Ученик самостоятельно определяет и исправляет ошибки в своём и чужом коде.
Умение использовать чужой код при разработке: анализировать и изменять.	Выполнение заданий на платформе.	Ученик самостоятельно может внести изменения в готовый код и использовать фрагмент готового кода в процессе реализации собственной программы.
Использование итерационного подхода.	Выполнение заданий на платформе.	Ученик пишет программы, разделяя код на отдельные части.
Личностные навыки		

Освоение социальной роли обучающегося и формирование личностного смысла учения.	Демонстрация связи между способностью выполнить интересную задачу и наличием/отсутствием соответствующих знаний.	За отведённое время ученик пытается не только выполнить базовые уровни, но и приступить к бонусным.
Развитие навыков сотрудничества со взрослыми и сверстниками и умения находить выходы из спорных ситуаций.	Использование программирования как способа показать преимущества работы в команде.	Ученик не боится просить помощь и сам пытается помогать одноклассникам и учителю.
Овладение начальными навыками адаптации в динамично изменяющемся и развивающемся мире.	Обсуждение научно-популярных материалов, демонстрирующих необходимость не просто овладения предметными навыками, но и развития умения учиться.	При работе над проектом ученик сначала пытается найти информацию в Интернете или теоретической справке на платформе и только при неудаче задаёт вопрос учителю.
Метапредметные навыки		
Развитие и формирование учебных действий.	Создание благоприятных условий для участия в диалоге, в коллективном обсуждении. Строится продуктивное взаимопонимание со сверстниками и взрослыми в процессе коллективной деятельности.	Ученик легко общается, не боится просить помощь или оказать её другим. Ученик способен кооперироваться, чтобы достичь цели.
Умение презентовать свою работу.	Презентация индивидуальных проектов. Учитель и другие ученики дают обратную связь. Учитель также даёт советы, каким образом это лучше делать.	Ученик во время презентации своих проектов пользуется вниманием аудитории.
Развитие критического и алгоритмического мышления учеников.	Выполнение логических операций: сравнение, анализ, синтез, обобщение, установление аналогий на уроке. Анализ любой получаемой информации, составление плана действий	Ученик строит логическую цепь рассуждений. Управляет своей деятельностью.

	для решения задачи.	
Развитие творческих способностей учеников.	Планирование и создание проекта по сценарию. Использование инструментов цифрового сервиса для воплощения своего проекта.	Ученик придумывает, обсуждает, планирует и реализует свой проект.

2-ой год обучения

Планируемые результаты	Способ достижения	Критерий достижения образовательного результата
Предметные навыки		
Формирование базовых навыков работы с основными инструментами разработчика на Python.	Изучение теоретического материала совместно с учителем, выполнение заданий на платформе с постепенным увеличением сложности.	Ученик оперирует базовыми профессиональными инструментами разработчика при работе над проектами.
Знакомство с библиотеками языка программирования Python: PyQt, Pygame, Pillow.	Изучение теоретического материала совместно с учителем, выполнение заданий в профессиональной среде с расширением образовательной платформы «Алгоритмики» с постепенным увеличением сложности.	Ученик самостоятельно пишет программы, используя библиотеки языка программирования Python: PyQt, Pygame, Pillow.
Развитие умений применять Python для решения задач практического характера и задач из смежных дисциплин.	Выполнение заданий в профессиональной среде с расширением образовательной платформы «Алгоритмики» с постепенным увеличением сложности.	Ученик самостоятельно пишет программы на языке Python, решающие задачи практического характера.
Знакомство с GitHub и его использование для публикации проектов.	Изучение технологии работы на GitHub и её применение на практике.	Ученик создаёт репозиторий со своим итоговым проектом и делится ссылкой на него.
Следование стандарту при оформлении кода.	Изучение теоретического материала совместно с учителем, выполнение	Ученик оформляет код в соответствии с общепринятыми

	заданий в профессиональной среде с расширением образовательной платформы «Алгоритмики».	стандартами и использует комментарии.
Умение тестировать программу и находить в ней ошибки.	Выполнение проектных заданий с постепенным увеличением сложности.	Ученик самостоятельно определяет и исправляет ошибки в своём и чужом коде.
Использование итерационного подхода.	Выполнение заданий в профессиональной среде с расширением образовательной платформы «Алгоритмики».	Ученик пишет программы, разделяя код на отдельные части.
Личностные навыки		
Освоение социальной роли обучающегося и формирование личностного смысла учения.	Демонстрация связи между способностью выполнить интересную задачу и наличием/отсутствием соответствующих знаний.	За отведённое время ученик пытается выполнить не только обязательную, но и дополнительную часть проектного задания.
Развитие навыков сотрудничества со взрослыми и сверстниками и умение находить выходы из спорных ситуаций.	Использование программирования как способа показать преимущества работы в команде.	Ученик не боится просить помощь и сам пытается помогать одноклассникам и учителю.
Овладение начальными навыками адаптации в динамично изменяющемся и развивающемся мире.	Обсуждение научно-популярных материалов, демонстрирующих необходимость не просто овладения предметными навыками, но и развития умения учиться.	При работе над проектом ученик сначала пытается найти информацию в теоретической справке на платформе и только при неудаче задаёт вопрос учителю.
Метапредметные навыки		
Развитие и формирование учебных действий.	Создание благоприятных условий для участия в диалоге, в коллективном обсуждении. Строится продуктивное взаимопонимание со сверстниками и	Ученик легко общается, не боится просить помощь или оказать её другим. Ученик способен кооперироваться, чтобы достичь цели.

	взрослыми в процессе коллективной деятельности.	
Умение презентовать свою работу.	Презентация индивидуальных проектов. Учитель и другие ученики дают обратную связь. Учитель также даёт советы, каким образом это лучше делать.	Ученик во время презентации своих проектов пользуется вниманием аудитории.
Развитие критического и алгоритмического мышления учеников.	Выполнение логических операций: сравнение, анализ, синтез, обобщение, установление аналогий на уроке. Анализ любой получаемой информации, составление плана действий для решения задачи.	Ученик строит логическую цепь рассуждений. Управляет своей деятельностью.
Развитие творческих способностей учеников.	Планирование и создание проекта по сценарию. Использование инструментов цифрового сервиса для воплощения своего проекта.	Ученик придумывает, обсуждает, планирует и реализует свой проект.

Формы аттестации и оценочные материалы. Текущий контроль сформированности результатов освоения программы осуществляется с помощью нескольких инструментов на нескольких уровнях:

- **на каждом занятии:** опрос, выполнение заданий на платформе, реализация проектов, самоконтроль ученика;
- **в конце каждого модуля:** проведение мини-хакатонов и презентации финальных проектов модуля и их оценка.

Для контроля сформированности результатов освоения программы с помощью выполнения заданий на платформе используются следующие механики:

- классический тест (выбор одного или нескольких правильных ответов),
- заполнение пропусков,
- классификация,
- сопоставление,
- сортировка.

На курсе возможны следующие **формы деятельности** учащихся: учебная дискуссия, занятие-проект, практическое занятие, конференция. При решении одной задачи ученики могут работать сначала в малой группе, затем индивидуально, а при проверке задачи – в паре с учителем.

Задания на платформе с автопроверкой (все задания на тему базовых понятий проверяются автоматически системой) направлены на формирование навыков написания программ на языке Python и умения читать код, вносить в него изменения (исправления) и

дополнять. Кроме того, предусмотрены ревью кода (выборочная оценка кода учеников на соблюдение стандартов его написания), конференции и круглые столы, на которых возможна взаимопроверка, чек-листы для преподавателей и учеников.

2. УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН.

1-ый год обучения

№ п/ п	Наименование раздела/темы	Количество часов		
		всего	теория	практика
<i>Модуль 1. Основы языка (8 часов).</i>				
1	Введение в Python	2	1	1
2.	Переменные	2	1	1
3.	Строки	2	1	1
4.	Вложенные конструкции	2	1	1
<i>Модуль 2. Управляющие конструкции (10 часов).</i>				
5.	Условный оператор	2	1	1
6.	Вложенный условный оператор	2	1	1
7.	Циклы	2	1	1
8.	Циклы. Продолжение	2	1	1
9.	Вложенные управляющие конструкции	2	1	1
<i>Модуль 3. Функции и модули (8 часов).</i>				
10.	Функции	2	1	1
11.	Функции. Продолжение	2	1	1
12.	Модули random и time	2	1	1
13.	Создание модулей	2	1	1
<i>Модуль 4. Модуль Turtle. Математика для разработчика (8 часов).</i>				
14.	Линейные алгоритмы	2	1	1
15.	Циклы	2	1	1
16.	Условный оператор	2	1	1
17.	Проект «Городская среда»	2	1	1
<i>Модуль 5. Объектно-ориентированное программирование (12 часов).</i>				
18.	Объекты и методы	2	1	1
19.	События	2	1	1
20.	Проект Simple Paint	2	1	1
21.	Классы	2	1	1
22.	Наследование	2	1	1
23.	Наследование. Продолжение	2	1	1
<i>Модуль 6. Основы разработки игр на PyGame (16 часов).</i>				
24.	Основы создания игр на Pygame	2	1	1
25.	Списки	2	1	1
26.	Игра Fast Clicker. Ч. 1	2	1	1
27.	Игра Fast Clicker. Ч. 2	2	1	1
28.	Игра Fast Clicker. Ч. 3	2	1	1

29.	Игра «Арканоид». Ч. 1	2	1	1
30.	Игра «Арканоид». Ч. 2	2	1	1
31.	Игра «Арканоид». Ч. 3	2	1	1
Модуль 7. Серия хакатонов (8 часов).				
32.	Воркшоп: навыки разработчика	2	1	1
33.	Хакатон. TestIT	2	1	1
34.	Хакатон. Simple Paint	2	1	1
35.	Хакатон Grow Up	2	1	1
Выпускной (2 часа).				
36.	Проектная работа. Промежуточная аттестация.	2	1	1
Всего:		72	36	36

2-ой год обучения

№ п/ п	Наименование раздела/темы	Количество часов		
		всего	теория	практика
Модуль 1. Структуры данных (8 часов).				
1	Повторение. Обработка исключений	2	1	1
2.	Повторение. Списки	2	1	1
3.	Словари	2	1	1
4.	Вложенные структуры данных	2	1	1
Модуль 2. Разработка оконных приложений (14 часов).				
5.	Классы. Введение в PyQt	4	2	2
6.	Проектирование интерфейса	2	1	1
7.	Создание оконного приложения Memory Card. Ч. 1	2	1	1
8.	Создание оконного приложения Memory Card. Ч. 2	2	1	1
9.	Создание оконного приложения Memory Card. Ч. 3	2	1	1
10.	Создание оконного приложения Memory Card. Ч. 4	2	1	1
Модуль 3. Работа с файлами (10 часов).				
11.	Основы работы с файлами	4	2	2
12.	Создание оконного приложения «Умные заметки». Ч. 1	2	1	1
13.	Создание оконного приложения «Умные заметки». Ч. 2	2	1	1
14.	Создание оконного приложения «Умные заметки». Ч. 3	2	1	1
Модуль 4. Автоматическая обработка изображений (10 часов).				
15.	Основы обработки изображений	4	2	2
16.	Приложение Easy Editor. Ч. 1	2	1	1
17.	Приложение Easy Editor. Ч. 2	2	1	1

18.	Приложение Easy Editor. Ч. 3	2	1	1
Модуль 5. Продвинутая разработка игр на PyGame (20 часов).				
19.	Основы создания игр	4	2	2
20.	Создание игры Лабиринт. Ч. 1	2	1	1
21.	Создание игры Лабиринт. Ч. 2	2	1	1
22.	Создание игры Лабиринт. Ч. 3	2	1	1
23.	Создание игры Шутер. Ч. 1	2	1	1
24.	Создание игры Шутер. Ч. 2	2	1	1
25.	Создание игры Шутер. Ч. 3	2	1	1
26.	Создание игры Шутер. Ч. 4	2	1	1
27.	Доработка и презентация проекта	2	1	1
Модуль 6. Публикация и распространение ПО (8 часов).				
28.	Сборка проекта в приложение	2	1	1
29.	Повторение. Введение в Git	2	1	1
30.	Создание игры Пинг-понг. Ч. 1	2	1	1
31.	Создание игры Пинг-понг. Ч. 2	2	1	1
Выпускной (2 часа).				
32.	Проектная работа. Итоговая аттестация.	2	1	1
Всего:		72	36	36
Итого:		144	72	72

3. СОДЕРЖАНИЕ ПРОГРАММЫ.

1-ый год обучения.

Модуль 1. Основы языка

1.1. Введение в Python. Функции ввода-вывода

Теория. Язык программирования Python. Понятия «язык программирования», «алгоритм», «программа». Знакомство с синтаксисом языка. Функция ввода `input()`. Функция вывода `print()`. Необходимость вывода данных.

Практика. Разработка программы, осуществляющей ввод-вывод данных, ввод команд с клавиатуры.

1.2. Переменные. Численные типы данных

Теория. Данные. Типы данных. Переменная. Числовой тип данных. Оператор присваивания. Приоритет операций. Выражение.

Практика. Разработка программ, осуществляющих обработку переменных числового типа.

1.3. Строки

Теория. Строковый тип данных. Операции со строками и числами. Изменение типов данных с помощью функций `int()` и `str()`.

Практика. Разработка программ, осуществляющих обработку строковых переменных (сумма строк, повтор строки n -раз, вычисление длины строки, получение символа строки по номеру, замена подстроки на другую подстроку, разбиение строки по разделителю).

1.4. Вложенные конструкции

Теория. Знакомство с вложенными конструкциями.

Практика. Разработка программ, содержащих элементарные вложенные конструкции.

Модуль 2. Управляющие конструкции

2.1. Логический тип данных. Условный оператор

Теория. Логический тип данных. Утверждение. Условие. Условный оператор. Синтаксис условного оператора. Операторы сравнения.

Практика. Разработка программ, содержащих условный оператор.

2.2. Вложенный условный оператор и условный оператор нескольких ветвей

Теория. Вложенный условный оператор. Уровни вложенности. Условный оператор нескольких ветвей.

Практика. Разработка программ, содержащих вложенный условный оператор.

2.3. Цикл while. Цикл со счётчиком

Теория. Цикл. Необходимость применения циклов. Типы циклов. Демонстрация синтаксиса цикла `while()`. Разбор примеров программ с использованием цикла. Цикл со счётчиком. Цикл с накоплением результата вычислений в переменной.

Практика. Разработка программ, содержащих циклы различных типов.

2.4. Вложенные алгоритмические конструкции (цикл в цикл, условный оператор в цикл и др.)

Теория. Вложенные алгоритмические конструкции. Цикл в цикле. Цикл с вложенными условными операторами. Целочисленное деление.

Практика. Разработка программ, осуществляющих обработку данных с помощью вложенных алгоритмических конструкций.

Модуль 3. Функции и модули

3.1. Создание функций. Локальная и глобальная области видимости

Теория. Функция. Локальная и глобальная функции. Оператор `def`. Параметры функции. Значение функции. Вызов функции. Синтаксис создания функций.

Практика. Разработка программ, осуществляющих обработку данных с помощью функций.

3.2. Использование одной функции внутри другой

Теория. Использование одной функции внутри другой.

Практика. Разработка программ, содержащих функцию внутри другой функции.

3.3. Модули. Использование встроенных модулей стандартной библиотеки

Теория. Модуль. Встроенные модули стандартной библиотеки. Модуль `random`.

Практика. Разработка программ, осуществляющих обработку данных с помощью модулей, встроенных в стандартную библиотеку.

3.4. Создание модулей. Подключение нескольких модулей

Теория. Создание модуля. Подключение модуля.

Практика. Разработка программ, осуществляющих обработку данных с помощью дополнительных модулей, не входящих в стандартную библиотеку.

Модуль 4. Модуль Turtle. Математика для разработчика

4.1. Графический исполнитель Turtle. Математика перемещения исполнителя

Теория. Исполнитель. Графический исполнитель. Пиксель. Координаты. Знакомство с модулем `Turtle`: рисование линий, поворот, ширина и цвет пера, перемещение пера в указанную точку. Математика перемещения исполнителя.

Практика. Разработка программ, управляющих исполнителем.

4.2. Условный оператор. Рисование простых геометрических фигур

Теория. Условный оператор в модуле Turtle. Рисование простых геометрических фигур.

Практика. Разработка программ, осуществляющих рисование простых геометрических фигур с использованием условного оператора.

4.3. Цикл. Рисование многоугольников

Теория. Циклы в модуле Turtle. Рисование многоугольников.

Практика. Разработка программ, осуществляющих рисование многоугольников.

4.4. Проектное занятие

Модуль 5. Объектно-ориентированное программирование

5.1. Объекты, их поля и методы

Теория. Объекты. Поля и методы объектов. Создание объектов.

Практика. Разработка программ, осуществляющих обработку данных с помощью передачи объекта в функцию.

5.2. Обработка событий мыши и клавиатуры

Теория. События. Обработка взаимодействия пользователя и программы. Обработка событий мыши и клавиатуры.

Практика. Разработка программ, осуществляющих обработку событий.

5.3. Проект Simple Paint

Теория. Планирование проекта Simple Paint.

Практика. Создание проекта Simple Paint.

5.4. Классы. Конструктор класса

Теория. Классы. Конструктор класса. Создание нестандартного объекта и описание его свойств и методов.

Практика. Разработка программ, осуществляющих обработку классов.

5.5. Наследование (от готового или своего класса)

Теория. Наследование. Класс-наследник. Использование готовых свойств и методов объекта. Добавление новых методов.

Практика. Разработка программ, использующих наследование.

Модуль 6. Основы разработки игр на PyGame

6.1. Основы разработки игр. Спрайт, игровой цикл

Теория. Спрайт. Игровой цикл: обработка событий, обновление игры, прорисовка, подсчёт времени. Библиотека play. Создание спрайтов разных типов и с разными свойствами. Расположение спрайта.

Практика. Разработка программ для создания спрайтов разных типов и управления ими.

6.2. Списки и методы работы с ними. Перебор списка в цикле for

Теория. Списки. Методы работы со списками. Перебор списков в цикле for.

Практика. Разработка программ, осуществляющих обработку списков.

6.3. Инструменты веб-аналитики

Теория. Знакомство с аналитикой и интерпретированием метрик.

Практика. Работа с аналитикой, интерпретирование метрик.

6.4. Игра Fast Clicker

Теория. Планирование игры Fast Clicker.

Практика. Создание игры Fast Clicker.

6.5. Игра «Аркиноид»

Теория. Планирование игры «Аркиноид».

Практика. Создание игры «Арканоид».

Модуль 7. Серия хакатонов

7.1. Воркшоп: навыки разработчика

Теория. Знакомство с хакатоном.

Практика. Использование полученных навыков в хакатонах. Применение soft и hard skills в различных жизненных ситуациях.

7.2. Хакатон

Теория. Разбор задания и порядка проведения хакатона.

Практика. Создание собственного проекта с использованием модулей, графического интерфейса и объектно-ориентированного программирования.

7.3. Выпускной

Теория. Планирование защиты проекта.

Практика. Защита проекта. Промежуточная аттестация.

2-ой год обучения.

Модуль 1. Структуры данных

1.1. Повторение. Обработка исключений

Теория. Функции ввода и вывода, переменные и типы данных. Алгоритмические конструкции — следование, ветвление и цикл. Модули Random и Time. Обработка исключений, операторы try, except.

Практика. Решение задач на повторение на платформе.

1.2. Повторение. Списки

Теория. Списки. Методы работы со списком: добавление, удаление, вставка, сортировка, подсчёт вхождений. Типы данных, которые может содержать список. Перебор элементов списка с помощью цикла for. Определение понятия «строка». Аналогии между методами работы со списками и со строками.

Практика. Решение задач на повторение на платформе.

1.3. Словари

Теория. Определение понятия «словарь». Добавление в словарь пары «ключ-значение». Методы работы с ключами и значениями словаря. Поиск ключа в словаре. Типы данных ключей и значений.

Практика. Решение задач на повторение на платформе.

1.4. Вложенные структуры данных

Теория. Правила создания лаконичного кода. Вложенные структуры. Применение методов к вложенным структурам. Обращение к объектам внутри вложенных структур с помощью индексов.

Практика. Решение задач повышенной сложности.

Модуль 2. Разработка оконных приложений

2.1. Классы. Введение в PyQt

Теория. Определение класса. Связь классов и объектов. Конструкторы, поля и методы в классах. Приложения и виджеты. Расположение виджетов.

Практика. Обработка событий средствами PyQt.

2.2. Проектирование интерфейса

Теория. Наследование. Суперклассы, классы-наследники, их методы. Виджеты QPushButton, QMessageBox и др.

Практика. Решение задач на повторение на платформе.

2.3. Последовательное создание профессионального (компьютерного) приложения Memory Card

Теория. Основные элементы приложения: окно, текст, кнопки, флажки, переключатели, списки, меню.

Практика. Проектирование интерфейса приложения с помощью лейаутов и виджетов.

Модуль 3. Работа с файлами

3.1. Основы работы с файлами

Теория. Атрибуты доступа к файлам. Оператор `with`. Чтение информации из файла. Строки и классы для считывания информации из файла.

Практика. Решение задач на повторение на платформе.

3.2. Создание оконного приложения «Умные заметки» с поиском записей по тегам

Теория. `Json`-файлы. Запись и чтение данных из `json`-файлов. Глобальные и локальные переменные. Программирование корректного отображения данных из файла.

Практика. Создание приложения для работы с заметками с возможностью создавать, удалять и редактировать заметки, задавать заметкам теги и выполнять поиск по ним.

Модуль 4. Автоматическая обработка изображений

4.1. Основы обработки изображений

Теория. Библиотека `Python PIL` как инструмент обработки изображений. Модули `Image` и `ImageFilter` библиотеки `PIL`. Параметры изображения `Image`, их изменение с помощью `PIL`.

Практика. Решение задач на повторение на платформе.

4.2. Создание приложения Easy Editor для обработки фотографий

Теория. Возможности и методы обработки изображений библиотекой `PIL`. Локальные и глобальные переменные. Программирование функций работы с файлами и интерфейсом. Массовая обработка изображений.

Практика. Создание приложения-фоторедактора `Easy Editor` (загрузка фото с компьютера, наложение ч/б фильтра, наведение резкости и поворот изображения).

Модуль 5. Продвинутое создание игр на PyGame

5.1. Основы создания игр

Теория. Обработка нажатия клавиши с помощью `PyGame`. Создание окна игры и спрайтов. Игровой цикл. Событие завершения игры.

Практика. Работа над собственным проектом.

5.2. Создание игры Лабиринт

Теория. Класс `Sprite`. Метод `super().__init__()`. Создание поля, метода и экземпляров собственного класса. Условия победы и поражения. Обработка внутриигровых событий и внешней среды.

Практика. Работа над собственным проектом.

5.3. Создание игры «Шутер»

Теория. Создание класса наследника. Задание спрайтов, управляемых с клавиатуры. Задание аудио- и визуального фона игры. Создание группы спрайтов и управление ей. Тест-кейсы и тестирование.

Практика. Работа над собственным проектом.

5.4. Доработка и презентация проекта

Теория. Доработка проекта. Презентация проекта. Приёмы успешных публичных выступлений. Google-презентации.

Практика. Работа над собственным проектом.

Модуль 6. Бонусный модуль — портфолио разработчика

6.1. Сборка проекта в приложение

Теория. Расширения. Система управления пакетов *pip*. Модуль *Puinstaller*. Упаковка игры в файл с расширением *.exe (.pkg)*.

Практика. Работа над собственным проектом.

6.2. Повторение. Введение в Git

Система контроля версий. Сервис GitHub. Понятия «репозиторий», «ветка», «коммит». Создание репозитория с проектом на *PuQt*.

Практика. Работа над собственным проектом.

6.3. Создание игры «Пинг-Понг»

Теория. Планирование проекта с помощью *mind map*. Реализация проекта по плану. Выбор лицензии на *GitHub* и заполнение файла *ReadMe*.

Практика. Работа над собственным проектом.

6.4. Выпускной

Теория. Планирование защиты проекта.

Практика. Защита проекта. Итоговая аттестация.

4. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ.

Материально-техническое обеспечение

Для каждого обучающегося:

- персональный ноутбук;
- персональный логин и пароль для доступа на платформу;
- платформа Алгоритмики с интеграцией *PuGame*, открывается через браузер *Google Chrome*;
- компьютеры (ноутбуки) должны быть подключены к единой сети с доступом в Интернет;
- Интернет (не менее 50 Мбит/сек);
- канцелярские принадлежности.

Для преподавателя:

- презентационное оборудование (проектор с экраном/телевизор с большим экраном) с возможностью подключения к компьютеру (ноутбуку);
- флипчарт с комплектом листов / маркерная доска, соответствующий набор письменных принадлежностей.

Требования к ПК:

- центральный процессор с архитектурой *x64* не старше 5 лет и не менее чем с 4 потоками;
- *Intel* серии *Core* (*Core i5, 7, 9*) не старше восьмого поколения, с частотой не менее 2,8 ГГц;
- *AMD* серии *FX* не ниже *FX-84350*, все модели серии *Ryzen*. Старшие модели серий *A10* и *A12* (*9700, 9800*), *Athlon 200GE*; *Apple M1, M2*;

- дискретная видеокарта с поддержкой графических редакторов DirectX 11/12 и новее, Shader Model 2.0 и новее;
- ОЗУ: DDR3 и новее, объёмом от 4 Гб (очень рекомендуем 8 Гб и больше);
- операционная система (ОС) Windows 10, 11 / MacOS 12 Monterey и выше / Linux 22.04 и выше;
- браузер Google Chrome или Яндекс.Браузер, последней версии обновления;
- колонки, микрофон, монитор диагональю не менее 15" с разрешением от 1920 × 1080;

Информационное обеспечение

1. Поурочные методические рекомендации к занятиям (платформа Алгоритмика);
2. Тематические презентации (платформа Алгоритмика);
3. Задания на платформе и в среде программирования с расширением учебной платформы (платформа Алгоритмика);
4. Справочный теоретический материал, размещённый на платформе (платформа Алгоритмика);
5. Бонусные задания на платформе (платформа Алгоритмика);
6. Дополнительные проектные задания;
7. Задачи со свободным решением для реализации в среде разработки на базе платформы «Алгоритмика».

Кадровое обеспечение

Уровень образования среднее профессиональное или высшее.

5. МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ПРОГРАММЫ

В отличие от других курсов программирования, Python Start предоставляет учащимся возможность думать как программисты, а не просто решать стандартный набор задач. Более того, проекты, выполняемые на уроках, охватывают разные области интересов, в том числе школьные предметы. В курсе используются механики peer-to-peer, позволяющие ученикам приобрести навыки взаимного обучения и взаимной проверки. Дети учатся самостоятельно находить и изучать новую информацию и доносить её до других, проводить ревью проектов и давать качественную обратную связь. Также ученики развивают умение самопрезентации и презентации своих проектов или идей как в группе, так и перед учителями и родителями.

Авторское воплощение замысла курса заключается в том, что ученики используют инструменты профессиональной разработки для реализации собственных проектов. Для этого сначала ученик получает базовые знания (алгоритмы, структурное программирование, использование библиотек в процедурном стиле), затем учится навыкам проектной деятельности, после чего реализует проекты в области своих увлечений. При этом базовая часть также подразумевает возможность выразить свои интересы, благодаря большому количеству задач разного уровня. Проекты выполняются преимущественно индивидуально, чтобы дети сначала научились работать над ними, затем уже – работать в группе. Эволюция компьютеров и программного обеспечения привела к достаточной простоте их освоения для самых неподготовленных пользователей, в том числе школьников.