

Центр цифрового образования детей «IT- куб»
(структурное подразделение АНО ДТ «Красноярский Кванториум»)

РЕКОМЕНДОВАНО
методическим советом

Протокол № 1
от «27» мая 2025 г.

УТВЕРЖДАЮ
Генеральный директор
Кениг С.Р.

Приказ № 15
от «27» мая 2025 г.



Дополнительная общеобразовательная общеразвивающая программа
технической направленности
углубленного уровня
«Проектная деятельность в IT»

Срок реализации:

1 год

Возраст детей:

13-17 лет

Составитель программы:

Нагуслаев Н.Т.

г. Красноярск, 2025 г.

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Данная программа разработана в соответствии с нормативными правовыми актами в области образования:

Федеральным законом от 29 декабря 2012 № 273-ФЗ «Об образовании в Российской Федерации»;

Указа Президента РФ от 07.05.2018 № 204 «О национальных целях и стратегических задачах развития Российской Федерации на период до 2024 года»;

Распоряжением Правительства РФ от 29.05.2015 № 996-р «Стратегия развития воспитания в Российской Федерации до 2025 года, утвержденная»;

Распоряжением Правительства РФ от 31.03.2022 № 678-р «Концепция развития дополнительного образования детей до 2030»;

Постановления Правительства Российской Федерации от 26.12.2017 № 1642 «Об утверждении государственной программы Российской Федерации «Развитие образования»;

Приказа Министерства просвещения РФ от 27.07.2022 №629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;

Постановления Главного государственного санитарного врача Российской Федерации от 28.09.2020 № 28 «Об утверждении санитарных правил СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи»;

Методическими рекомендациями по проектированию дополнительных общеразвивающих программ (включая разноуровневые программы) (Приложение к письму Министерства образования и науки Российской Федерации от 18.11.2015 г. № 09-3242);

Методическими рекомендациями по реализации дополнительных общеобразовательных программ с применением электронного обучения и дистанционных образовательных технологий (Приложение к письму Министерства просвещения РФ от 31.01.2022 г. № 1ДГ 245/06);

Методическими рекомендациями «Об использовании государственных символов Российской Федерации при обучении и воспитании детей и молодежи в образовательных организациях, а также организациях отдыха детей и их оздоровления» (Письмо Министерства просвещения РФ от 15.04.2022 № СК-295/06);

Методическими рекомендациями по созданию и функционированию центров цифрового образования «IT-куб» (Письмо Министерства просвещения РФ от 10.11.2021 №ТВ-1984/04.

Дополнительная общеобразовательная общеразвивающая программа «Проектная деятельность в IT» технической направленности, для обучающихся от 13-17 лет, объем годовой нагрузки 144 часа, направлена на

приобретение базовых навыков проектной деятельности и разработку собственных проектов.

Для освоения данной программы необходимы знания, полученные при изучении программы «Программирование на языке Python 2» или «Мобильная разработка-2».

Сегодня важным звеном, а также ожидаемым карьерным развитием специалиста в любой компании и особенно в IT-сфере является деятельность по управлению проектами или «проектный менеджмент». Успех любого проекта в IT, будь то создание веб-сайта, интернет-магазина, робота или запуск нового мобильного приложения, зависит во многом от менеджера проекта (руководителя проектной команды), который наряду с глубокими знаниями в области технологий и программирования, должен владеть навыками, методами и инструментами управления проектом и его ресурсами.

Знания, полученные в результате изучения программы «Проектная деятельность», помогут ученикам освоить базовые понятия в управлении проектами, а также на практике применить теоретические знания для решения собственных проектных задач.

Дополнительная общеобразовательная общеразвивающая программа «Проектная деятельность» (далее - программа) имеет техническую направленность, средний уровень сложности и ориентирована на обучающихся 13-17 лет. Программа в объеме 144 часа рассчитана на один год из расчета 4 часа в неделю.

1.1. Актуальность программы

Большинство курсов и образовательных программ по программированию нацелены только на изучение языка программирования. Данная программа нацелена на изучение программирования с точки зрения разработки конкретного проекта, и позволит обучающимся увидеть более полную картину работы программиста и разработки проекта.

Изучение основ управления IT-проектами способствуют формированию современного мировоззрения, развитию интеллектуальных способностей, развивает способности к самообразованию, умению ставить задачи, планировать их выполнение, умению разбивать решение задачи на подзадачи, способность работать с информацией: находить, оценивать и использовать информацию из различных источников. Обучение по данной программе дает обучающимся возможность реализовать собственные проекты.

Осознание уникальности и многогранности разработки проектов в сфере IT может стать мотивацией учеников к погружению в сферу профессиональной деятельности, а освоение ее современных подходов и инструментов станет выгодной инвестицией в их будущее, ведь решение любой жизненной задачи можно рассматривать как проект, а методологию управления проектами считать одним из самых эффективных способов успешного достижения поставленной цели.

Данная программа позволяет обучающимся пройти весь жизненный цикл IT-проекта от идеи до запуска продукта и получить необходимый опыт разработки.

Высокую вовлеченность обучающихся обеспечивает самостоятельный выбор темы проекта.

Программа рассчитана на четыре этапа обучения:

- 1) Закрепление базовых навыков
- 2) Управление проектами
- 3) Работа с Git и клиент-серверной разработкой
- 4) Разработка проекта

Данная программа формирует учебные компетенции, которые позволят обучающимся в будущем создавать собственные программы, игры, сайты. Программа направлена на развитие у детей навыков по управлению проектами, совершенствование навыков программирования и разработки ПО, представления о профессии программиста и менеджера по проектам.

1.2. Отличительные особенности

Отличительной особенностью данной образовательной программы является то, что процесс обучения выстроен с учетом используемых современных образовательных технологий. Программа предполагает использование и реализацию таких форм организации материала, которые допускают освоение специализированных знаний и языка программирования, обеспечивают трансляцию общей и целостной картины в рамках содержательно-тематического направления программ. Итоговое задание позволяет проявить творческий подход к решению задачи.

Основной целью программы является создание условий для осознания учениками важности построения дальнейшей индивидуальной образовательной траектории и ранней профориентации через знакомство с перспективными направлениями развития IT-индустрии, спектром IT-профессий и возможностями профессиональной самореализации, спецификой деятельности различных специалистов.

Предметом и главной темой обсуждения в этой программе станет управление проектами - деятельность, направленная на достижение поставленных задач, реализацию определённых планов, используя имеющиеся ресурсы - время, капитал, людей, материалы и инструменты; минимизацию рисков и отклонений от плана, эффективного управления изменениями.

В ходе программы важно изменить сложившийся у большинства людей стереотип: чтобы состояться как специалист в IT-индустрии недостаточно научиться хорошо программировать.

Важно показать, что возрастает значение профессиональных компетенций коллективной разработки программного обеспечения, знание современных направлений, методов и технологий управления IT-проектами:

- 1) понимание обязанностей различных участников команды по разработке программного обеспечения: руководитель разработки программного обеспечения, руководитель технической группы (team leader), архитектор, программист, тестировщик, дизайнер, верстальщик, аналитик;
- 2) владение современными стратегиями и технологиями организации коллективной разработки программного обеспечения, включая системы управления версиями, процессы непрерывной интеграции, стандарты оформления кода и методы инспекции кода;
- 3) понимание основных направлений развития методов коллективной разработки, их отличий и целесообразности применения в зависимости от типа решаемых задач и требований организации;
- 4) владение методологиями разработки программных продуктов.

1.3. Адресат общеразвивающей программы

Набор на программу осуществляется в соответствии с Порядком организации образовательной деятельности по дополнительным общеобразовательным общеразвивающим программам автономной некоммерческой организации «Красноярский детский технопарк «Кванториум».

Дополнительная общеобразовательная общеразвивающая программа «Проектная деятельность» предназначена для детей в возрасте 13–17 лет, успешно освоивших ДООП «Программирование на языке Python 2» или «Мобильная разработка-2».

В связи с ориентированностью программы на практическую индивидуальную (групповую) работу максимальное количество обучающихся в группе не должно превышать 8 человек.

Образовательный процесс в разновозрастных учебных группах выстраивается на идеях педагогики сотрудничества: учение без принуждения, самоанализа, создания благоприятного интеллектуального фона учебной группы, личностного подхода, продвижения в индивидуальном темпе, самоконтроля и взаимоконтроля. Реализация положений педагогики сотрудничества эффективно воплощается в жизнь при применении диалогических форм обучения, которые подразумевают творческое отношение и обмен креативной деятельностью. Осуществление педагогического диалога в учебном процессе позволяет в ходе учебно-познавательной деятельности детей развивать их коллективистские связи.

Возрастные особенности группы:

13-14 лет – подростковый возраст. Характерная особенность - индивидуальное самосознание, сознательное выражение индивидуальности. Основное стремление - самоутверждение. В подростковом возрасте интересы детей стабилизируются. Основное новообразование - формирование взрослости как желания жить во взрослом обществе. На социально-нравственном уровне - потребность иметь и отстаивать собственные мнения и

оценки; на интеллектуально-энергетическом уровне - потребность овладеть элементами саморазвития и понять сферу интересов подростка; на культурном уровне - потребность отразить взрослость во внешности и поведении. Основными критериями являются следующие. Роль преподавателей высших учебных заведений, работающих с подростками, заключается в проведении регулярной подготовки к введению их в социально значимые группы.

15-17 лет – подростковый возраст. Завершение физической и психической зрелости. Социальная готовность к общественно полезному производительному труду и гражданской ответственности. В отличие от подросткового возраста, где индивидуальность проявляется через самоопределение - "кто я", подростковая индивидуальность выражается через самовыражение - "как я влияю". Основная задача педагога дополнительного образования, занимающегося с детьми 15-17 лет, - подготовить их к полноценной социальной жизни и разрешить противоречие между подготовкой к полноценной социальной жизни и недопущением задержек в содержании и организации учебной деятельности.

Основной формой обучения являются практические занятия, так как дети этой возрастной группы обладают внутренней уравновешенностью и готовностью к активному участию в практической деятельности. Они также заинтересованы в общественной и групповой деятельности, так как возрастает значение групп, обмена опытом, взаимоотношений со сверстниками и оценки их действий, слов и поступков сверстниками и старшими. Дети стремятся завоевать авторитет в собственных глазах и занять ценное положение в группе. Поэтому в программе предусмотрены практические упражнения в соревновании личностей, которые позволяют каждому проявить себя и найти свое место в группе.

Следует также отметить, что для детей в этом возрасте характерны такие познавательные процессы, как изменение структуры личности и возникновение интересов к ней, развитие абстрактных форм мышления, формирование более осознанной и целенаправленной деятельности, проявление стремления к независимости и самостоятельности, формирование самооценки. Эти процессы формируют раннее профессиональное самоопределение учащегося.

1.4. Педагогическая целесообразность

Над кодом мобильных приложений, игр, корпоративных информационных систем, созданием программно-аппаратных комплексов, цифровых гаджетов, web-сервисов и т.п. работает команда высококлассных профессионалов различной специализации: программисты и инженеры, дизайнеры и тестировщики и т.п., которые превращают идеи в реальные продукты, необходимые миллионам людей. Такими сложными командами необходимо уметь управлять. Поэтому важным звеном в деятельности любой IT-компании является управление проектами или «проектный менеджмент».

В термин «управление проектами» разные люди вкладывают разный смысл, зачастую не имея четкого представления о моделях, методах, инструментах и технологиях управления проектами. Между тем, управление проектами – это профессиональная деятельность, основанная на использовании современных научных знаний, навыков, методов, средств и технологий и ориентированная на эффективное получение желаемых результатов.

Применение методологии управления проектами дает возможность четко определить цели и результаты проекта, дать им количественные характеристики, временные, стоимостные и качественные параметры проекта, создать четкий план проекта, выделить, оценить риски и предотвратить возможные негативные последствия во время реализации проекта.

На сегодняшний день методология управления проектами доказала свое право считаться одним из самых эффективных способов успешной реализации поставленных целей.

Проектная деятельность, связанная с непрерывным улучшением и изменением функциональности ИТ-продуктов, ближе к изобретательству и творчеству, чем к ремеслу, поэтому не всегда хорошо прогнозируется. Это в свою очередь, служит причиной сложности организации процесса взаимодействия между всеми заинтересованными сторонами: заказчиками, инвесторами, участниками команды разработчиков, пользователями, увеличению количества связей и точек их пересечения, что объективно вызывает трудности в принятии решений, ведет к ошибкам в продукте, превышению стоимости разработки, затягиванию выполнения графиков работ.

Кроме того, важный фактор разработки программного обеспечения – потенциально огромное количество пользователей продукта труда разработчиков. Компьютерные программы разрабатываются для людей (заказчиков/пользователей) и, следовательно, всегда сопровождается постоянным социальным взаимодействием не только внутри команды, но и с людьми «снаружи».

Социальный аспект в разработке программного обеспечения – это очень важная составляющая многих ИТ-проектов, которая отличает их от проектов создания материальных вещей.

Осознание сложности и уникальности управления ИТ-проектами, в свою очередь, может служить и мотивацией учеников к погружению именно в эту сферу профессиональной деятельности, овладению искусством укладываться в сроки, бюджет проекта, рационально использовать ресурсы проекта, организовывать работу команды специалистов - с одной стороны; прогнозировать, каким должен быть и будет продукт, насколько он будет востребован - с другой.

В программе применены методики реальной разработки ИТ-проекта, что для традиционного обучения не свойственно. Программа не

ограничивается разработкой только проектной идеи, она рассчитана на создание конкурентоспособного продукта.

1.5 Срок реализации программы и объем учебных часов

Программа рассчитана на 1 год обучения. Годовая нагрузка на обучающегося составляет 144 часа.

1.6. Формы обучения, виды занятий и режим занятий

Учебные занятия проходят в очной форме. Режим занятий – 2 раза в неделю по 2 академических часа (1 академический час - 40 минут) с обязательным перерывом, что определяется Санитарно-эпидемиологическими правилами и нормативами СанПиН 2.4.3648-20.

При проведении занятий используются комбинированные занятия – изложение нового материала, проверка пройденного материала, закрепление полученных знаний, самостоятельная работа.

При проведении занятий используются следующие формы работы:

- 1) демонстрационная, когда обучающиеся слушают объяснения педагога и наблюдают за демонстрационным экраном или экранами компьютеров на ученических рабочих местах;
- 2) фронтальная, когда обучающиеся синхронно работают под управлением педагога;
- 3) самостоятельная, когда обучающиеся выполняют индивидуальные задания в течение части занятия;
- 4) повторение и усвоение пройденного материала осуществляется через контрольные и проверочные работы, анализ полученных результатов;
- 5) закрепление знаний, умений и навыков через постановку задачи и самостоятельную работу обучающегося под руководством педагога;
- 6) применение полученных знаний и навыков через прикладную работу обучающегося, использующего на практике приобретенные компетенции;
- 7) мозговые штурмы для генерации идей и гипотез по проекту;
- 8) параллельная разработка, когда обучающиеся вместе работают над одним проектом, но разрабатывают разный функционал;
- 9) разработка в режиме спринтов, когда обучающиеся разрабатывают определенный объем работы, решают подзадачу в короткий интервал времени;
- 10) стендапы, когда обучающиеся собираются и отчитываются по результатам спринта, чтобы обсудить ход выполнения работы. Цель такого совещания заключается в выявлении проблем, которые могут повлиять на достижение цели спринта.

1.7. Цель и задачи программы

Цель программы:

Сформировать у обучающихся комплексные навыки разработки программного обеспечения и управления IT-проектами, обеспечивающие их готовность к самостоятельной работе в профессиональной среде.

Задачи программы:

1. Сформировать базовые представления о проектной деятельности
 - Изучить ключевые понятия проектного подхода.
 - Разобрать примеры успешных IT-проектов и их жизненный цикл.
2. Ознакомить со спецификой IT-проектов
 - Рассмотреть особенности разработки программного обеспечения.
 - Изучить различия между коммерческими, учебными и open-source проектами.
3. Освоить методы управления проектами
 - Изучить основные методологии (Agile, Scrum, Kanban, Waterfall).
 - Научиться применять инструменты планирования (Jira, GitHub Projects).
4. Развить навыки командной работы
 - Отработать распределение ролей в проекте (разработчик, тестировщик, менеджер).
 - Научиться использовать системы контроля версий (Git, GitHub/GitLab) для совместной работы.
5. Усовершенствовать технические навыки программирования
 - Научиться разрабатывать, тестировать и отлаживать программы на Python.
 - Освоить принципы клиент-серверной разработки.
6. Сформировать самостоятельность и креативность
 - Развить умение самостоятельно ставить задачи и находить решения.
 - Поощрять нестандартные подходы к реализации проектов.
7. Научиться работать с технической документацией
 - Освоить чтение и применение документации к языкам программирования и библиотекам.
 - Научиться оформлять проектную документацию (README, техническое задание).
8. Расширить профессиональный кругозор
 - Познакомиться с актуальными тенденциями в IT (искусственный интеллект, DevOps, облачные технологии).
 - Изучить примеры современных технологических стеков.

1.8. Планируемые результаты освоения программы

Личностное и профессиональное самоопределение (мотивация к получению профессий в наукоемких областях через интерес к организационно-управленческому аспекту в IT-индустрии) - один из важнейших планируемых образовательных результатов.

Поэтому, знакомя учеников с современным методами управления IT-проектами, которые призваны способствовать развитию экономики, улучшить качество жизни и работы людей, ускорить и изменить процессы и способы решения различных задач, следует обратить внимание обучающихся на то, что процессы управления IT-проектами характеризуются многоплановостью и многофункциональностью, требуют от специалистов дополнительных навыков и компетенций, многие из которых связаны как с умением программировать, так и с навыками планирования, принятия решений и продуктивной коммуникации, высокой степенью личной ответственности и профессиональной исполнительности. Освоение начальных форм познавательной и личной рефлексии - одна из основных задач.

Отличительной особенностью образовательной программы является ориентация на кооперированную деятельность учеников, на развитие у них коммуникативных универсальных учебных действий:

- 1) умение организовывать продуктивное сотрудничество и совместную деятельность с одноклассниками;
- 2) работать индивидуально и в группе: находить общее решение и разрешать конфликты на основе согласования позиций и учёта интересов;
- 3) формулировать, аргументировать и отстаивать своё мнение.

Обучающиеся учатся работать самостоятельно и в команде, вступать в диалог и вести его, выполнять разные роли и обязанности. Приобретаемый ими социальный опыт и навыки практически значимы.

Учащиеся будут знать:

- 1) возможности современных и перспективных средств разработки программных продуктов, технических средств;
- 2) методологии разработки программного обеспечения и технологии программирования;
- 3) методологии управления проектами.

Учащиеся будут уметь:

- 1) вырабатывать варианты реализации программного обеспечения;
- 2) проводить оценку и обоснование выбранных решений;
- 3) разрабатывать алгоритм задачи;
- 4) работать в сети интернет для поиска информации;
- 5) работать в команде;
- 6) управлять собственными проектами.

Личностные результаты:

- 1) Учащиеся приобретут навыки самостоятельной организации своей деятельности, саморазвития и самовоспитания.

- 2) У обучающихся формируется готовность и способность к самостоятельной, творческой деятельности, к образованию, в том числе самообразованию; готовность к осознанному выбору будущей профессии.

Метапредметные результаты:

- 1) умение планировать;
- 2) умение анализировать;
- 3) умение алгоритмизировать;
- 4) умение организовывать свой труд и труд команды.

Предметные результаты:

- 1) сформированы базовые представления о проектах и особенностях проектного подхода к организации деятельности на различных примерах
- 2) сформировано представление о специфике IT-проектов и их основных особенностях;
- 3) сформировано представление о методах управления проектами;
- 4) развиты навыки коллективной разработки;
- 5) усовершенствованы навыки разработки, тестирования и отладки несложных программ;
- 6) сформированы навыки работы с документацией к языку программирования и программным продуктам;
- 7) Получен опыт проектной деятельности, создания, редактирования, оформления, сохранения, передачи информационных объектов различного типа с помощью современных программных средств.

1.9. Формы подведения итогов обучения

Текущий Контроль качества образования осуществляется в форме тестов, выполнения практических и проектных работ.

По итогам каждого этапа проводится итоговая аттестация в форме представления части проектной работы в форме презентации и демонстрации программной части.

Промежуточный контроль – защита проектной идеи, проводится через полгода после начала занятий. Итоговый контроль освоения программы проводится защиту индивидуального или группового проекта в рамках региональной научно-технической конференции красноярского технопарка «Кванториум» (далее – НТК). В результате проекты получают экспертную оценку. В ней принимает участие педагогический состав АНО ДТ «Красноярский Кванториум» и представители организаций-партнеров.

Ознакомиться с предлагаемыми вариантами проектов для итогового кейса можно в Приложении 1.

С критериями экспертной оценки индивидуального проекта можно ознакомиться в Приложении 2. Количество баллов, набранных за итоговую

аттестацию, переводится в один из уровней освоения образовательной программы согласно таблице:

Уровень освоения	Баллы
Низкий	0 – 49
Средний	50 – 99
Высокий	Более 90

2. УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН

№	Наименование разделов	Количество академических часов			Форма контроля
		Всего часов	Теория	Практика	
1	Закрепление базовых навыков	20	5	15	
1.1	Вычисления	2	1	1	—
1.2	Условия	4	1	3	Выполнение контрольного задания
1.3	Циклы	4	1	3	Выполнение контрольного задания
1.4	Списки, кортежи, множества, словари	4	1	3	Выполнение контрольного задания
1.5	Функции, библиотеки	6	1	5	—
2	Управление проектами	34	13	21	
2.1	Методологии управления проектом	6	2	4	—
2.2	Как правильно определить тему проекта	2	1	1	—
2.3	Формулирование актуальности проекта	2	1	1	—
2.4	Поиск проблемы	2	1	1	—
2.5	Формулировка цели	2	1	1	—
2.6	Формулировка задач	2	1	1	—
2.7	Поиск аналогов. Анализ	2	1	1	—
2.8	Проработка технического решения	4	1	3	—
2.9	Изучение технологий производства	4	2	2	—
2.10	Составление плана работы	2	1	1	—
2.11	Оформление презентации	4	1	3	—
2.12	Промежуточная аттестация. Предзащита проектной идеи	2	0	2	Защита идеи проекта
3	Работа с Git и клиент-серверной разработкой	72	28	44	
3.1	Введение в Git	4	2	2	—
3.2	Основные команды Git	4	2	2	—

3.3	Ветвление и слияние в Git	6	2	4	—
3.4	Работа с удаленными репозиториями (GitHub, GitLab)	4	2	2	—
3.5	Введение в клиент-серверную разработку	4	2	2	—
3.6	Основы Flask: настройка и запуск	6	2	4	—
3.7	Создание REST API на Flask	6	2	4	—
3.8	Работа с базами данных в Flask	6	2	4	—
3.9	Аутентификация и авторизация в Flask	6	2	4	—
3.10	Разработка клиента для Web	8	3	5	—
3.11	Разработка клиента для Android	8	3	5	—
3.12	Тестирование и отладка приложений	6	2	4	—
3.13	Деплой приложений	4	2	2	Практическое задание
4	Разработка проекта	18	2	18	
4.1	Создание проекта	14	2	12	—
4.2	Подготовка презентации	2	0	2	—
4.3	Итоговая аттестация	2	0	2	Защита проекта
Итог		144	56	88	

3. СОДЕРЖАНИЕ ПРОГРАММЫ

1.1 Тема: Вычисления

Теоретическая работа: Простые типы данных: целый, вещественный, символьный, логический и их представление в ПК. Приведение типов. Работа с переменными и константами. Использование математических функций. Организация ввода и вывода данных.

Практическая работа: Работа с простыми типами данных: целым, вещественным, символьным и логическим. Организация ввода и вывода данных.

1.2 Тема: Условия

Теоретическая работа: Конструкция ветвления. Разветвляющиеся программы. Условные выражения. Операции отношения. Логические операции. Полная и сокращенная форма условного оператора. Блоки и составные операторы. Оператор множественного выбора.

Практическая работа: Работа с логическим типом данных. Объявление и использование в программе переменных, констант и выражений логического типа. Использование простых и составных условий. Полная и сокращенная формы условного оператора. Создание разветвляющейся программы с помощью условных операторов. Оператор множественного выбора. Организация множественного разветвления в программе. Выполнение контрольного задания (Приложение 3).

1.3 Тема: Циклы

Теоретическая работа: Циклы. Циклические программы. Понятия итерации, счетчика и параметров цикла. Циклов со счетчиком. Циклы с предусловием.

Практическая работа: повтор команд. Переменные цикла. Операторы while. Оператор for. Бесконечный цикл. Выход из цикла. Выполнение контрольного задания (Приложение 4).

1.4 Тема: Списки, кортежи, множества, словари

Теоретическая работа: Основы множеств и алгебры логики. Круги Эйлера. списки. Индексация и обратная индексация, схожесть со строками. Отличия кортежей от списков. Практическое использование списков.

Практическая работа: Работа типами данных: Кортежи, tuple().Списки. Организация ввода и вывода данных. Срезы. Встроенные методы. Выполнение контрольного задания (Приложение 5).

1.5 Тема: Функции, библиотеки

Теоретическая работа: Создание функций с переменными аргументами. Лямбда-функции. Подходы к программированию. Как правильно разбивать программу на функции. Вывод из функции. Аргументы.

Практическая работа: Создание функций. Аргументы. Правильный вывод и ввод данных. создание глобальных и локальных переменных и структур данных. Вызов глобальных переменных и ограничения локальных. функция return. Присваивание функций.

2.1 Тема: Методологии управления проектом

Теоретическая работа: Что такое проект и командный проект. Отличия проекта от другой командной работы. Основные методологии управления проектом. Гибкие методологии. Scrum. Agile. Водопад. MVP.

Практическая работа: Работа на тренажере. Жизнь как проект. Разбор популярных сервисов на MVP и последующие версии.

2.2 Тема: Как правильно определить тему проекта

Теоретическая работа: Откуда берется тема проекта. Проблематика. Виды проектов и зависимость темы проекта от вида.

Практическая работа: Распределение темы на виды проектов. Составление тем готовых проектов.

2.3 Тема: Формулирование актуальности проекта.

Теоретическая работа: Обоснование актуальности проекта. Примеры актуальности проекта.

Практическая работа: Определение актуальности тестовых проектов.

2.4 Тема: Поиск проблемы

Теоретическая работа: Проблемное интервью. HADI-циклы. Анализ проблемы

Практическая работа: Определение основных проблем, которые существуют в конкретной ситуации.

2.5 Тема: Формулировка цели.

Теоретическая работа: На какой вопрос должна отвечать цель. Примеры целей проектов.

Практическая работа: Определение цели тестовых проектов.

2.6 Тема: Формулировка задач.

Теоретическая работа: На какой вопрос должна отвечать задача. Примеры задач проектов. Отличия цели от задачи и их связь.

Практическая работа: Определение задач к целям тестовых проектов.

2.7 Тема: Поиск аналогов. Анализ.

Теоретическая работа: Понятие аналога и анализа. Поиск аналогов и методы анализа информации.

Практическая работа: Построение сравнительной таблицы аналогов.

2.8 Тема: Проработка технического решения.

Теоретическая работа: Сферы разработки проектов. Виды проектов в IT. Технические особенности каждого вида.

Практическая работа: Определение сферы проекта. Определение технического решения для конкретного проекта. Особенности реализации.

2.9 Тема: Изучение технологий производства.

Теоретическая работа: Определение сложности проекта. Роли в проектах разных видов. Распределение компетенций.

Практическая работа: Распределение ролей в зависимости от компетенций. Определение сложности конкретного проекта.

2.10 Тема: Составление плана работы.

Теоретическая работа: Основные шаги для составления плана работы.

Практическая работа: Примерный календарный план. Объем работ. Распределение ролей.

2.11 Тема: Оформление презентации.

Теоретическая работа: Структура презентации. Сценарий. Содержание. Подача. Дизайн.

Практическая работа: Разработка сценария презентации. Разработка презентации в tilda или power point. Разработка и отработка выступления.

2.12 Промежуточная аттестация

Практическая работа: Предзащита проектной идеи. Обсуждение проектов.

3.1 Тема: Введение в Git

Теоретическая работа: Основные понятия Git: репозиторий, коммит, ветка. Преимущества использования системы контроля версий.

Практическая работа: Установка Git, создание первого репозитория, выполнение базовых команд (git init, git add, git commit).

3.2 Тема: Основные команды Git

Теоретическая работа: Работа с историей коммитов, отмена изменений, игнорирование файлов (.gitignore).

Практическая работа: Выполнение команд git log, git checkout, git reset, git revert.

3.3 Тема: Ветвление и слияние в Git

Теоретическая работа: Стратегии ветвления, разрешение конфликтов.

Практическая работа: Создание и переключение между ветками, слияние веток, разрешение конфликтов.

3.4 Тема: Работа с удаленными репозиториями

Теоретическая работа: Платформы для хостинга Git-репозиториях (GitHub, GitLab).

Практическая работа: Клонирование репозитория, работа с git pull, git push, создание pull request.

3.5 Тема: Введение в клиент-серверную разработку

Теоретическая работа: Основы HTTP, REST API, клиент-серверная архитектура.

Практическая работа: Написание простого HTTP-сервера на Flask.

3.6 Тема: Основы Flask: настройка и запуск

Теоретическая работа: Структура Flask-приложения, маршрутизация.

Практическая работа: Создание базового Flask-приложения, настройка маршрутов.

3.7 Тема: Создание REST API на Flask

Теоретическая работа: Методы HTTP, статус-коды, сериализация данных.

Практическая работа: Разработка REST API с использованием Flask-RESTful.

3.8 Тема: Работа с базами данных в Flask

Теоретическая работа: Подключение и работа с SQLAlchemy.

Практическая работа: Создание моделей, выполнение CRUD-операций.

3.9 Тема: Аутентификация и авторизация в Flask

Теоретическая работа: Механизмы аутентификации (JWT, OAuth).

Практическая работа: Реализация аутентификации в Flask-приложении.

3.10 Тема: Разработка клиента для Web

Теоретическая работа: Основы фронтенд-разработки (HTML, CSS, JavaScript).

Практическая работа: Создание веб-интерфейса для взаимодействия с API.

3.11 Тема: Разработка клиента для Android

Теоретическая работа: Основы разработки под Android (Kotlin/Java).

Практическая работа: Создание мобильного приложения для взаимодействия с API.

3.12 Тема: Тестирование и отладка приложений

Теоретическая работа: Виды тестирования (юнит-тесты, интеграционные тесты).

Практическая работа: Написание тестов для Flask-приложения и клиентов.

3.13 Тема: Деплой приложений

Теоретическая работа: Платформы для деплоя (Heroku, AWS).

Практическая работа: Развертывание сервера и клиентов в облаке.

4.1 Тема: Создание проекта

Теоретическая работа: Распределение ролей в команде (бэкенд, фронтенд, мобильная разработка, тестирование). Инструменты для совместной работы (Jira, Git).

Практическая работа: Разработка проекта по выбранной теме.

4.2 Тема: Подготовка презентации

Практическая работа: Оформление презентации проекта.

4.3 Тема: Итоговая аттестация

Практическая работа: Защита проекта.

4. СПИСОК ЛИТЕРАТУРЫ

1. Гаско, Рик. Простой Python просто с нуля [Электронный ресурс] / Рик Гаско. – М.: СОЛОН-Пресс, 2019.
2. Златопольский, Д.М. Программирование: типовые задачи, алгоритмы, методы [Электронный ресурс] / Д.М. Златопольский. – М.: БИНОМ. Лаборатория знаний, 2007.
3. Златопольский, Д. М. Сборник задач по программированию [Электронный ресурс] / Д.М. Златопольский. – 3-е изд., перераб. и доп. – СПб.: БХВ-Петербург, 2011.
4. Меньшиков, Ф.В. Олимпиадные задачи по программированию [Электронный ресурс] / Ф.В. Меньшиков. – СПб.: Питер, 2006.
5. Мюллер, Джон Пол. Python для чайников [Электронный ресурс] / Джон Пол Мюллер. - 2-е изд.: Пер с англ. – СПб.: ООО «Диалектика», 2019.
6. Окулов, С.М. Основы программирования: учебное пособие [Электронный ресурс] / С.М. Окулов. – 10-е изд., электрон. – Москва : Лаборатория знаний, 2020.
7. Прохоренок, Н.А. Python 3. Самое необходимое [Электронный ресурс] / Н. А. Прохоренок, В. А. Дронов. – 2-е изд., перераб. и доп. – СПб.: БХВ-Петербург, 2019.
8. Пупышев, В.В. 128 задач по началам программирования [Электронный ресурс] / В.В. Пупышев. – М.: БИНОМ. Лаборатория знаний, 2009.
9. Пэйн, Брайсон. Python для детей и родителей [Электронный ресурс] / Брайсон Пейн [пер. с англ. М.А. Райтмана]. - М.: Издательство «Э», 2017.
10. Свейгарт, Эл. Учим Python, делая крутые игры [Электронный ресурс] / Эл. Свейгарт. - М: Эксмо, 2018.
11. Седер, Наоми. Python. Экспресс-курс [Электронный ресурс] / Наоми Седер. - 3-е изд. – СПб.: Питер, 2019.
12. Столяров, А.В. Оформление программного кода [Электронный ресурс] / А.В. Столяров. – 2-е изд., испр. и доп. – Москва: МАКС Пресс, 2019.
13. Шень, А. Программирование: теоремы и задачи [Электронный ресурс] / А. Шень. - 6-е изд., дополненное. М.: МЦНМО, 2017.
14. Шуман, Х.Г. Python для детей [Электронный ресурс] / Х.Г. Шуман [пер. с нем. М.А. Райтман]. – М.: ДМК Пресс, 2019.
15. <https://www.python.org>
16. <https://learn.javascript.ru>
17. <https://developer.mozilla.org/ru/>
18. <https://www.tensorflow.org/>

5. ИНФОРМАЦИОННО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ И МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОСНАЩЕНИЕ ПРОГРАММЫ

5.1 Материально-техническое обеспечение программы

№ п/п	Наименование оборудования (ФПО)	Примерная модель (РВПО)	Единица измерения	Количество
1	"Презентационное оборудование"			
1.1	Напольная мобильная стойка для интерактивных досок или универсальное настенное крепление	ONKRON TS1330	шт	1
1.2	Моноблочное интерактивное устройство	SMART SBID-MX265-V2	шт	1
2	"Дополнительное оборудование"			
2.1	Комплект кабелей и переходников	Atcom High speed HDMI - HDMI MOST Lite LRG FAZA FOP-05GS-500	шт	1
2.2	Флипчарт магнитно-маркерный на треноге	Attache	шт	1
3	"Профильное оборудование"			
3.1	WEB-камера (рабочее место педагога)	logitech C922	шт	1
3.2	Наушники (рабочее место обучающегося)	SVEN AP-320M/321M	шт	12
3.3	Ноутбук тип 2 (рабочее место обучающегося)	Acer TravelMate P2 TMP215-52	шт	12
3.4	Ноутбук тип 1 (рабочее место педагога)	Acer TravelMate P2 TMP215-52	шт	1

5.2. Информационное обеспечение программы

Книги по Python:

- «Изучаем Python», Марк Лутц — книга крайне подробно рассматривает общие концепции языка, всё написано понятно и доступно.
- «Высокопроизводительный Python: практическое пособие для людей», Миша Горелик, Ян Освальд — в книге рассказывается о многопоточности, асинхронном и мультипроцессорном исполнении, о том, как организовать кластерные вычисления и как потреблять меньше оперативной памяти.
- «Flask Web Development», Мигель Гринберг — автор этой книги, описал собственный процесс разработки веб-приложений при помощи Flask. Все его примеры кода — части единого приложения. В самом начале вы создаете его в базовой форме и с каждой новой главой расширяете.

Telegram каналы:

- Python etc. Здесь вы найдете практические советы по написанию кода на Python (с отрывками-примерами). Канал англоязычный.
- PythonistPro. Англоязычный канал по Python. Тут вас ждут ссылки на статьи, видео и книги, а также посты с приемами программирования.

Подпишитесь на канал — и будете не только Python учить, но и английский подтягивать.

- Python Academy. Очень популярный канал (почти 49 тысяч человек) с советами по использованию Python. Каждый пост — коротенький рассказ о каком-нибудь модуле или функции, или приеме. У канала есть свой чат, но есть и обсуждение в комментариях под постами.

Youtube каналы:

- Хауди Хо™ – Просто о мире IT! – уроки по программированию, в том числе по
- Гоша Дударь – разные тематические видео на IT тематику: Python в
- Иван Викторович – мини-курс Python с нуля и разработка игр на Pygame.

Предлагаемые варианты проектов

Итоговым результатом освоения обучающимися полученных в процессе обучения навыков и компетенций в рамках представленной программы является итоговая защита проекта в рамках НТК.

В приложении приведены предлагаемые варианты проектов, которые может выполнить ученик. Ученик должен самостоятельно предложить тему проекта и согласовать её с преподавателем. Свободный выбор темы, позволяет заинтересовать ученика. Так как каждый ученик сможет выбрать тему в зависимости от своих знаний и интересов. Работа над персональным проектом — предполагает построение цели, задач для ее достижения, и выполнение по персональному ТЗ.

Предлагаемые варианты проектов:

- Создание чат-бота;
- Создание сайта;
- Создание мобильного приложения;
- Создание десктопного при.

Критерии оценки итогового проекта

№ п/п	Объект оценки	Критерии	Баллы
1.	Оценка созданного изделия (проведенного исследования, разработанного digital проекта, проекта в области промышленного дизайна)	Новизна и актуальность темы проекта	от 0 до 10
		Привлекательность и оригинальность (внешнего вида созданного изделия, проведенного исследования, разработанного digital проекта, проекта в области промышленного дизайна)	от 0 до 10
		Работоспособность и (или) возможность для внедрения (изготовленного изделия, проведенного исследования, разработанного digital проекта, проекта в области промышленного дизайна)	от 0 до 10
		Качество (изготовленного изделия, проведения исследования, разработанного digital проекта, проекта в области промышленного дизайна)	от 0 до 10
		Перспективность и конкурентоспособность (созданного изделия, проведенного исследования, разработанного digital проекта, проекта в области промышленного дизайна)	от 0 до 10
2.	Оценка описания проекта	Формулировка темы, цели и задач проекта	от 0 до 7
		Исследование проблемы проекта	от 0 до 7
		Соответствие результата проекта поставленной цели	от 0 до 7
		Исследование целевой группы (аудитории)	от 0 до 7
		Обоснование экономической составляющей (создания изделия, проведения исследования, разработки digital проекта, создания проекта в области промышленного дизайна)	от 0 до 7
		Анализ аналогов проекта	от 0 до 7
		Дальнейшее развитие проекта	от 0 до 7
3.	Оценка защиты проекта	Соблюдение регламента публичной защиты	от 0 до 5
		Качество подачи материала и представления	от 0 до 5
		Понимание сути задаваемых вопросов и аргументированность ответов докладчика	от 0 до 5
		Качество презентации и презентационных материалов	от 0 до 5
4.	Премияльные баллы от эксперта		от 0 до 10

Контрольные задания по теме «Условия»

1. Пользователь вводит трёхзначное число. Найдите сумму его цифр.

Пример

Ввод:

241

Вывод:

7

2. Пользователь вводит три целых числа. Два из них равны друг другу. Найти третье число, не равное остальным. Если среди введенных чисел не оказалось двух равных друг другу или все числа равны, выведите строку "Ошибка".

Пример

Ввод:

4

3

4

Вывод:

3

3. Пользователь вводит целое положительное число. Проверьте является ли это число трехзначным, если является, то выведите строку "YES", иначе "NO".

Пример

Ввод:

454

Вывод:

YES

4. Пользователь вводит номер текущего месяца. Программа должна вывести текущее время года.

Пример

Ввод:

12

Вывод:

Зима

5. Пользователь вводит 3 числа. Найти и вывести наименьшее из чисел (не использовать функцию min).

Пример

Ввод:

4

3

6

Вывод:

3

Контрольные задания по теме «Циклы»

1. Пользователь последовательно вводит числа. Если пользователь введёт 0, программа должна закончить свою работу и вывести на экран "СТОП!".

Пример

Ввод:

5

4

0

Вывод:

СТОП!

2. Пользователь последовательно вводит числа. Если пользователь введёт 0, программа должна закончить свою работу и вывести на экран сумму всех чётных чисел (0 не считается частью последовательности).

Пример

Ввод:

5

4

6

0

Вывод:

10

3. Пользователь последовательно вводит числа. Если пользователь введёт 0, программа должна закончить свою работу и вывести на экран максимальное число кратное трём (0 не считается частью последовательности).

Пример

Ввод:

5

4

6

9

0

Вывод:

9

4. Пользователь последовательно вводит числа. Если пользователь введёт 0, программа должна закончить свою работу и вывести на экран минимальное положительное число (0 не считается частью последовательности).

Пример

Ввод:

5

-4

8

3

-1

0

Вывод:

3

5. Пользователь вводит число N, затем N целых чисел. Найти количество отрицательных среди введенных чисел.

Пример

Ввод:

5

-1

6

-4

7

-3

Вывод:

3

6. Пользователь вводит число N, затем N целых чисел. Найти сумму всех положительных чисел.

Пример

Ввод:

3

-1

4

7

Вывод:

11

7. Пользователь вводит число N, затем N положительных целых чисел. Найти сумму всех трёхзначных чисел.

Пример

Ввод:

4

12

433

7122

100

Вывод:

533

8. Пользователь вводит число N, затем N целых чисел. Найти минимальное из чисел.

Пример

Ввод:

3

1

4

7

Вывод:

1

Контрольные задания по теме «Списки, кортежи, множества, словари»

1. Создать список из 8 случайных чисел в диапазоне от -10 до 10 включительно. Найти максимально чётное число.

Пример

Список:

4 8 -3 9 5 -4 -2 1

Вывод:

8

2. Пользователь вводит строку из чисел, указанных через пробел. Найти сумму чисел, кратных 5.

Пример

Ввод:

4 6 3 5 10

Вывод:

15

3. Пользователь вводит число N, затем N целых чисел. Найти количество чисел, заканчивающихся на 4.

Пример

Ввод:

4

34

11

44

73

Вывод:

2

4. Программа получает на вход последовательность, каждое число записано в отдельной строке. Последовательность завершается числом 0, при считывании которого программа должна закончить свою работу. Найти минимальное трёхзначное число.

Пример

Ввод:

64

455

3421

435

0

Вывод:

435

5. Создать список из 8 случайных чисел в диапазоне от -10 до 10 включительно. Удалить из списка все отрицательные элементы.

Пример

Ввод:

-4 6 3 -5 2 7 10 -3

Вывод:

6 3 2 7 10

6. Создать список из 8 случайных чисел в диапазоне от -10 до 10 включительно. Заменить элементы, которые больше обоих своих соседей, на максимальный элемент списка (крайние элементы не учитывать).

Пример

Ввод:

-4 6 3 -5 2 7 10 -3

Вывод:

-4 10 3 -5 2 7 10 -3

