

Экспертное заключение

Об итогах экспертизы программы по информатике района  
для 10-11 класса Тервомайского

На основании Закона «Об образовании Кыргызской Республики», положения Управления образования мэрии г. Бишкек от 08.01.2013 г. экспертная комиссия Управления образования провела экспертизу программы Авторская программа по ул. предмету «Информатика» 10-11 классов «Основы программирования на Python»  
название программы

учителя Кемачевой Е.А.  
Ф.И.О учителя

УР АФКШЛ №1  
название школы

Представленная на экспертизу программа является Авторская рабочая программа для школ аулук-классного изучения программирования для 10-11 классов «Основы программирования на Python» разработана и составлена в соответствии с Законом КР «Об образовании». Настоящая программа определяет общие компетенции, результаты и содержание основ программирования 10-11 классов. Представленная программа соответствует требованиям, предъявляемым к изучению информатики в 10-11 классе штануской физико-математической школы с профильным изучением предмете

На основании проведенной «12» февраля 2020 г. экспертизы экспертная комиссия предлагает:

1. Признать программу по информатике 10-11 классов «Основы программирования на Python» авторскую  
учителя Кемачевой Е.А. УР АФКШЛ №1  
Ф.И.О учителя название школы

соответствующей установленным требованиям к организации обучения по эксперименту.

2. УР АФКШЛ №1 г. Бишкек Тервомайского района имеет

право ведения образовательной деятельности по реализации программы по учебному курсу «информатика» для 10-11 классов «Основы программирования на Python», составленной учителем

Кемачевой Еленой Александровной  
название программы Ф.И.О учителя

Председатель комиссии: Алексеева Г.Б. Хансу  
Члены комиссии: Тимонова Р.К.  
Виноградов Д.В.  
Мамышев Б.А.  
Кемачева Е.А.

Учебный комплекс авторской физико–математической школы лицея № 61 Якира Е.Б.

**Программа  
по учебному предмету «Информатика»  
для школ с углублённым изучением  
программирования  
«Основы программирования на PYTHON»  
для учащихся 10 – 11 классов**

Программу разработали:  
учитель ШМО информатики  
УК АФМШЛ № 61  
Клепачёва Е.А.

Тип программы: экспериментальная  
Программа утверждена:  
на заседании Научно-методического  
совета  
Протокол №1 от 30.09.20 г.

г. Бишкек – 2019 год

## Оглавление

I. Аннотация к программе .....	4
II. Пояснительная записка .....	4
III. Цели и задачи программы: .....	4
IV. Методологические основы предмета .....	7
V. Место учебного предмета в учебном плане.....	10
VI. Общая характеристика учебного предмета .....	10
VII. Содержательные линии. Логика построения курса.....	14
VIII. Основное содержание учебного курса для 10 класса, Образовательные результаты и оценивание процесса .....	15
IX. Основное содержание учебного курса для 11 класса.....	17
X. Образовательные результаты и оценивание процесса. Основные стратегии оценивания достижений учащихся .....	19
XI. Нормы отметок.....	22
XII. Список использованной литературы: .....	24

## **I. Аннотация к программе**

Данная программа "Основы программирования на Python" представляет собой вводный курс по программированию, дающий представление о базовых понятиях структурного программирования (данных, переменных, ветвлениях, циклах и функциях) в 10 классе и объектно-ориентированном программировании - в 11 классе.

ЯП Python – это язык, обладающий рядом преимуществ перед другими языками для начинающих изучать программирование (ясность кода, быстрота реализации).

Алгоритмизация и программирование является центральным, но наиболее сложным разделом в курсе информатики основного общего образования. Важные аспекты данной темы изучаются поверхностно (на базовом уровне 5-9 классы), а количество часов отводимых на прохождение учебной программы является недостаточным для полноценной подготовки обучающихся к предметным олимпиадам и конкурсам по программированию. Данный курс, рассчитанный на 2 года обучения позволяет восполнить потребности в дополнительной подготовке по предмету как мотивированных и одаренных обучающихся так и учащихся с низкой мотивацией.

## **II. Пояснительная записка**

Авторская рабочая программа для школ с углубленным изучением программирования «Информатика 10-11 классы» разработана и составлена в соответствии с Законом Кыргызской Республики (ред.г.Бишкек, от 30 апреля 2003 года N 92) «Об образовании» (с изм. и доп. от 8 июня 2017 года № 100)

Настоящая программа определяет общие компетентности, результаты и содержание основ программирования 10-11 класс. В программе учитываются возрастные и психологические особенности школьников, учитываются межпредметные связи. Программа является ориентиром для составления календарно-тематических и поурочных планов. Она предоставляет широкие возможности для реализации различных подходов к построению учебного курса с учётом позиции и творческого потенциала учителя, индивидуальных способностей, интересов и потребностей учащихся, материальной базы, национальных традиций и особенностей школы, где предмет Информатика, наряду с физикой и математикой является профильным предметом. Настоящая программа нацелена на применение в процессе обучения коммуникативно-деятельностного подхода для того, чтобы обеспечить формирование коммуникативных умений, в частности умения задавать вопросы, путем организации коллективной практической деятельности, направленной на постановку и решение учащимися конкретных учебных задач.

## **III. Цели и задачи программы:**

*Основной целью изучения основ программирования учащимися является - формирование у учащихся компетенции алгоритмического мышления, умения составлять алгоритмы и реализовывать их в выбранной среде программирования, формирование представлений об основных этапах разработки программного обеспечения от постановки задачи до составления программы, её ввода в память компьютера, тестирования и отладки, формирование умений использовать интегрированную среду разработки для составления программ и выполнять составленные программы.*

### Задачи рабочей программы:

Изучение информатики в старших классах (10-11) вносит значительный вклад в достижение главных целей профильного образования и способствует решению следующих задач:

#### 10 классы

- *развитию общеучебных умений и навыков на основе средств и методов программирования*, а именно формирование понятий, которые обеспечивают метапредметные представления о месте информатики и ее методов среди других научных отраслей (алгоритм, структуры данных, модуль, объект, модель, функция, процедура, класс, компиляция, отладка, коллекция, массив); самостоятельно планировать и осуществлять индивидуальную и коллективную информационную деятельность, представлять и оценивать ее результаты;
- *формированию у учащихся умения организации собственной учебной деятельности*, включающими: целеполагание как постановку учебной задачи на основе соотнесения того, что уже известно, и того, что требуется установить; планирование - определение последовательности промежуточных целей с учетом конечного результата, разбиение задачи на подзадачи, разработка последовательности и структуры действий, необходимых для достижения цели при помощи фиксированного набора средств; прогнозирование - предвосхищение результата; контроль - интерпретация полученного результата, его соотнесение с имеющимися данными с целью установления соответствия или несоответствия (обнаружения ошибки); коррекция - внесение необходимых дополнений и корректив в план действий в случае обнаружения ошибки; оценка - осознание учащимся того, насколько качественно им решена учебно-познавательная задача;  
– *воспитанию ответственного и избирательного отношения к информации*; развитию познавательных, интеллектуальных и творческих способностей учащихся;

#### 11 классы

- *формированию целостного мировоззрения*, соответствующего современному уровню развития науки и общественной практики за счет развития представлений о программах их не только применение, но и создание как важнейшем стратегическом ресурсе развития личности, общества; понимания роли информационных процессов в современном мире;
- *формированию у учащихся умения организации собственной учебной деятельности*, включающими: целеполагание как постановку учебной задачи на основе соотнесения того, что уже известно, и того, что требуется установить; планирование - определение последовательности промежуточных целей с учетом конечного результата, разбиение задачи на подзадачи, разработка последовательности и структуры действий, необходимых для достижения цели при помощи фиксированного набора средств; прогнозирование - предвосхищение результата; контроль - интерпретация полученного результата, его соотнесение с имеющимися данными с целью установления соответствия или несоответствия (обнаружения ошибки); коррекция - внесение необходимых дополнений и корректив в план действий в случае обнаружения ошибки; оценка - осознание учащимся того, насколько качественно им решена учебно-познавательная задача;
- *формированию у учащихся компетенций информационного моделирования* как основного

метода приобретения знаний; обеспечение социализации обучающихся в современном информационном обществе; подготовка к будущей профессиональной деятельности.

- *формирование у учащихся основных универсальных компетенций* информационного характера;
- *формирование у учащихся широкого спектра компетенций*: использования средств информационных и коммуникационных технологий для сбора, хранения, преобразования и передачи различных видов информации; овладения способами и методами освоения новых инструментальных средств;
- *формирование у учащихся основных компетенций* самостоятельной работы, первичные умения и навыки исследовательской деятельности, принятия решений и управления объектами с помощью составленных для них алгоритмов;
- *формирование у учащихся компетенций* продуктивного взаимодействия и сотрудничества со сверстниками и взрослыми: умения правильно, четко и однозначно формулировать мысль в понятной собеседнику форме; умения работы в группе; умения выступать перед аудиторией, представляя ей результаты своей работы с помощью средств ИКТ.
- *совершенствованию общеучебных и общекультурных навыков работы с информацией* в процессе систематизации и обобщения имеющихся знаний, умений и способов деятельности в области информатики и ИКТ; развитию навыков самостоятельной учебной деятельности школьников (учебного проектирования, моделирования, исследовательской деятельности и т. д.);
- *воспитанию ответственного и избирательного отношения к информации* с учетом правовых и этических аспектов ее распространения, воспитанию стремления к продолжению образования и созидательной деятельности с применением средств ИКТ.

## IV. Методологические основы предмета

### *Коммуникативно-деятельностный подход в изучении предмета «Основы программирования на Python»*

Методологической основой предмета является компетентностный и системно-деятельностный подход, в рамках которого реализуются современные стратегии обучения, предполагающие использование информатики в процессе изучения всех школьных предметов, во внеурочной и внешкольной деятельности на протяжении всего периода обучения в школе.

Развитие **коммуникативной деятельности** учащихся при обучении информатике обеспечивает социальную компетентность и учет позиции других людей, умение слушать и вступать в диалог, участвовать в коллективном обсуждении проблем, интегрироваться в группы для решения проектных задач. В рамках **деятельностного подхода** ученики должны научиться применять академические знания в решении практических задач на основе использования реальных систем, использования ИКТ в процессе изучения всех предметов, во внеурочной и внешкольной деятельности на протяжении всего периода обучения в школе.

Для реализации поставленных задач используются две формы обучения:

- урочная форма, в которой учитель объясняет новый материал и консультирует обучающихся в процессе выполнения ими практических заданий или проектных работ, совместно с учащимися включается в поиск методов решения поставленных задач, предлагает и обсуждает с ними достоинства и недостатки различных подходов и способов решения;
- дистанционная, в которой учащиеся получают информацию и задания по сети на предметном сайте (<http://olymp.krsu.edu.kg>) и/или на других образовательных ресурсах сети Интернет, участвуют в дистанционных олимпиадах и конкурсах.

Основным методом обучения в данном курсе является метод проектов. Проектная деятельность позволяет развить исследовательские и творческие способности учащихся, умение работать в команде и планировать свои действия, умение работать с различными источниками информации и анализировать собранный материал. В основу проектной деятельности положен цикл решения задачи:

1. постановка задачи;
2. построение и анализ моделей;
3. выбор алгоритма решения;
4. формализация;
5. программная реализация;
6. анализ полученных результатов;
7. использование полученных результатов.

В процессе решения задач формируется язык, общий для многих научных областей.

Разработка каждого проекта реализуется в форме выполнения практической работы (решения олимпиадного задания, составления прикладной программы) на компьютере. Кроме выполнения проектов, учащимся предлагаются практические задания для самостоятельного выполнения (мини-проекты).

Недостаточность того или иного подхода, приводит:

- при отсутствии деятельностного подхода: к излишней теоретизации и неумению использовать ИКТ в обычной жизни;
- при отсутствии коммуникативного подхода: к неумению задавать вопросы, находить и обсуждать решения проблем с другими;

- при отсутствии проектного или соревновательного (олимпиады, конкурсы) подхода: к отсутствию творческого и креативного подхода к решению задач, что делает процесс обучения скучным, а значит менее эффективным.

### ***Принципы обучения и отбора учебных ресурсов***

Дидактические требования к содержанию образования по информатике направлены на формирование двух типов знаний и умений учащихся:

- *алгоритмического* - развитие логического и алгоритмического мышления;
- *технологического* - формирование умений работы с прикладным программным обеспечением для решения различных практических задач.

В связи с этим, основными дидактическими принципами обучения и отбора учебных ресурсов в рамках предмета информатика являются:

- ***научности*** - обеспечение достаточной глубины, корректности и научной достоверности содержания учебного материала, с учетом последних достижений в науке и технике;
- ***доступности*** - определение степени теоретической сложности учебного материала в соответствии с возрастными и индивидуальными особенностями учащихся. Недопустимость необоснованного усложнения и увеличения объема учебного материала, при которых овладение этим материалом становится непосильным для учащихся;
- ***проблемности*** - предъявление материала в проблемном виде, когда учащийся сталкивается с учебной проблемной ситуацией, требующей разрешения, при этом его мыслительная активность возрастает;
- ***наглядности*** - учет чувственного восприятия изучаемых объектов, их макетов либо моделей и их наблюдение учащимися. Требование обеспечения наглядности при использовании компьютерных технологий реализуется на принципиально новом качественном уровне;
- ***самостоятельности*** - самостоятельность и активизация деятельности предполагает обеспечение учащихся электронными средствами обучения, позволяющими развивать у учащихся самостоятельность по поиску и отбору необходимой учебной информации при четком понимании конечных целей и задач учебной деятельности, а также осуществлять выбор той либо иной траектории обучения и управления ходом событий;
- ***систематичности и последовательности*** - обеспечение последовательного усвоения учащимися определенных знаний в рамках изучаемого учебного предмета, формирование знаний и умений учащихся в определенной системе, в строго логическом порядке и применение их учащимися в учебной и практической деятельности.
- ***прочности усвоения знаний*** - обеспечение возможности глубокого осмысления учащимися учебного материала;
- ***взаимосвязанности и взаимообусловленности*** смежных предметов;
- ***эстетического аспекта*** содержания образования.

Кроме традиционных дидактических требований к содержанию образования по информатике предъявляются и **специфические дидактические требования**, обусловленные использованием преимуществ современных информационных и телекоммуникационных технологий:

- ***адаптивности*** - адаптируемость содержания образования к индивидуальным возможностям учащихся;
- ***интерактивности*** - в содержании образования должно иметь место взаимодействие

обучающегося с программным средством;

- **реализации возможностей компьютерной визуализации учебной информации** - использование современных средств отображения информации и возможностей современного программного обеспечения для визуализации информации;
- **развития интеллектуального потенциала обучающегося** - формирование разнообразных стилей мышления: алгоритмического, наглядно-образного, рефлексивного, теоретического, умения принимать рациональные или вариативные решения в различных ситуациях, умений по обработке различных видов информации на основе применения ИКТ;
- **полноты (целостности) и непрерывности дидактического цикла обучения** - предоставление возможности выполнения определенной задачи в пределах одного сеанса работы.

## V. Место учебного предмета в учебном плане.

Рабочая программа рассчитана на 2 года в 10 классе– 3 часа в неделю (102 часа в учебном году), в 11 классе - 2 часа в неделю (68 часов в учебном году), в том числе:

- 10 класс – 1 час лекция, 2 ч. – практические занятия
- 11 класс – 2 часа и лекции и практики.

## VI. Общая характеристика учебного предмета

Информатика – это естественнонаучная дисциплина о закономерностях протекания информационных процессов в системах различной природы, а также о методах и средствах их автоматизации.

Многие положения, развиваемые информатикой, рассматриваются как основа создания и использования информационных и коммуникационных технологий — одного из наиболее значимых технологических достижений современной цивилизации. Вместе с математикой, физикой, химией, биологией курс информатики закладывает основы естественнонаучного мировоззрения.

### Компетентности

При изучении информатики в школе у учащихся формируются *ключевые* и *предметные* компетентности.

*Ключевые компетентности* являются образовательным результатом, формируемым и реализуемым на содержании конкретных предметов и базирующимся на социальном опыте учащегося, а также характеризуются многофункциональностью.

В соответствии с категориями ресурсов, которые используются человеком в личной и профессиональной сферах (информационные ресурсы, другие люди и группы людей, личностные качества и возможности самого человека), ключевыми являются следующие компетентности:

1) *информационная компетентность* - готовность использовать информацию для планирования и осуществления своей деятельности, формирования аргументированных выводов. Предполагает умение работать с информацией: целенаправленно искать недостающую информацию, сопоставлять отдельные фрагменты, владеть навыками целостного анализа и постановки гипотез; позволяет человеку принимать осознанные решения на основе критически осмысленной информации;

2) *социально-коммуникативная компетентность* - готовность соотносить свои устремления с интересами других людей и социальных групп, цивилизованно отстаивать свою точку зрения на основе признания разнообразия позиций и уважительного отношения к ценностям (религиозным, этническим, профессиональным, личностным) других людей. Готовность получать в диалоге необходимую информацию и представлять ее в устной и письменной формах для разрешения личностных, социальных и профессиональных проблем. Позволяет использовать ресурсы других людей и социальных институтов для решения задач;

3) *самоорганизация и разрешение проблем* - готовность обнаруживать противоречия в информации, учебной и жизненной ситуациях и разрешать их, используя разнообразные способы, самостоятельно или во взаимодействии с другими людьми, а также принимать решения о дальнейших действиях.

Основанием для определения уровней сформированности компетентностей является степень самостоятельности учащегося и сложность использованных видов деятельности при решении задач.

Выделяются три уровня сформированности ключевых компетентностей:

1) первый уровень (репродуктивный) характеризуется умением учащихся следовать образцу (заданному алгоритму выполнения действия);

2) второй уровень (продуктивный) характеризуется способностью выполнять простую по составу деятельность, применять усвоенный алгоритм деятельности в другой ситуации;

3) третий уровень (креативный) подразумевает осуществление сложносоставной деятельности с элементами самостоятельного ее конструирования, обоснования и применения коммуникационных технологий.

Выделенные уровни компетентностей не сопоставлены непосредственно со ступенями обучения. На каждой ступени обучения предполагается овладение учащимся компетентностями всех уровней в соответствии с его возрастными и индивидуальными особенностями и с учетом образовательных условий.

В соответствии с тремя компетентностями в Государственном образовательном стандарте определены специальные (предметные) компетентности – частные по отношению к ключевым компетентностям, которые формируются в рамках учебного предмета.

**Ключевые компетентности (информационная, социально-коммуникативная, самоорганизация и разрешение проблем) реализуются через следующие предметные компетентности:**

- **Пользовательская компетентность** включает в себя навыки владения различными программными средствами, сервисами интернета, цифровой техникой;

- **Компетентность в области программирования** овладевает основными понятиями логики, математики и построения алгоритмов, а также навыками проектирования программного обеспечения.

- **Коммуникативно-информационная компетентность** включает в себя навыки работы в команде для решения различных задач, получение и анализ информации через различные каналы, безопасность работы с информацией.

Владение пользовательской компетентностью означает, что ученик владеет навыками пользователя ИКТ, т.е.:

- соблюдает технику безопасности при работе с различными устройствами, оценивает риски для здоровья, связанные с использованием цифровых средств (начиная с эргономических аспектов и заканчивая технологической зависимостью);

- различает основные и периферийные устройства, а также их строение и принципы работы;

- имеет представление о назначении системного и прикладного программного обеспечения и умеет ими пользоваться;

- исходя из своей цели, создает цифровой контент на различных платформах и в различных средах (тексты, фото, аудио, видео);

- понимает базовые принципы работы Интернета (сайты, доменные имена, протоколы, и т.д.);

- осуществляет поиск информации в Интернете с помощью поисковых сервисов, применяя для этого различные методы поиска информации: поиск по ключевым словам, сортировку, фильтры;

- владеет основными принципами работы с Интернет-сервисами (э-почта, облачные хранилища, социальные сети, и др.);

Владение компетентностью в области программирования означает, что ученик владеет алгоритмическим мышлением и навыками разработки программ, т.е.:

- понимает, что такое алгоритмы, умеет различать алгоритмы в повседневной жизни, умеет представлять алгоритм (план действий) для формального исполнителя с помощью команд на формальном языке;

- умеет устанавливать и работать в средах разработки, предназначенных для разработки программного обеспечения, записывать команды с последующим их исполнением (вводом и выводом данных);

- составляет простейшие программы с помощью языка программирования, содержащие переменные, циклы, условные операторы и функции;

- умеет писать программы для обработки данных (списки, таблицы или массивы), текстовой информации, создания графики;

- умеет анализировать процессы, так чтобы решить, какую часть процесса можно автоматизировать, решать проблемы путем их разбиения на более мелкие задачи (декомпозиция);

- разрабатывает программы с использованием двух или более языков программирования (как минимум, один из них должен быть формальным, а второй может быть визуальным языком программирования);

- умеет проектировать и создавать интернет-сайты;

- умеет разрабатывать простейшие программы для управления роботом, или интерактивную игру.

Владение коммуникативно-информационной компетентностью предполагает, что ученик, используя информацию и ИКТ, умеет критически оценивать и решать собственные реальные жизненные задачи, в том числе:

- умеет определить свои потребности в информации и находит подходящую информацию – в частности, для саморазвития, учебы, поведения в обществе и проблемных ситуациях, а также для проведения исследовательской работы;

- умеет искать и делиться, анализировать и проверять достоверность информации, полученной из различных источников, в том числе и интернет;

- умеет критически оценивать адекватность, надежность и целостность найденной информации, сравнивать заданные источники информации в интернете с точки зрения их пригодности, объективности, взвешенности и адекватности.

- умеет защищать свои цифровые средства, принимая меры безопасности (например, с помощью антивирусных программ и программ, нацеленных против других вредоносных программ);

- различает уровни безопасности цифровых сред (например, http или https, сертификаты безопасности);

- умеет надлежащим образом оформить творческую работу, учитывая общепринятую практику авторского права в отношении контента, созданного как им самим, так и другими, различает закрытое и открытое ПО, виды свободных лицензий;

- безопасно и этично использует свою цифровую идентичность и осторожен при цифровом общении с посторонними посредством интернет-сред.

- умеет творчески и целенаправленно использовать возможности цифровых технологий при решении жизненных проблем и при повышении эффективности своей учебы.

Информатика имеет большое и все возрастающее число *междисциплинарных связей*, причем как на уровне понятийного аппарата, так и на уровне инструментария. Многие предметные знания и способы деятельности (включая использование средств ИКТ), освоенные обучающимися на базе информатики, находят применение как в рамках образовательного процесса при изучении других предметных областей, так и в иных жизненных ситуациях,

становятся значимыми для формирования качеств личности, т. е. ориентированы на формирование метапредметных и личностных результатов. На протяжении всего периода становления школьной информатики в ней накапливался опыт формирования образовательных результатов, которые в настоящее время принято называть современными образовательными результатами.

Одной из основных черт нашего времени является всевозрастающая изменчивость окружающего мира. В этих условиях велика роль фундаментального образования, обеспечивающего профессиональную мобильность человека, готовность его к освоению новых технологий, в том числе, информационных. Необходимость подготовки личности к быстро наступающим переменам в обществе требует развития разнообразных форм мышления, формирования у учащихся умений организации собственной учебной деятельности, их ориентации на деятельностную жизненную позицию.

Экспериментальная рабочая программа делится на 2 этапа: 10 класс – алгоритмический (структурный) ЯП Python; 11 класс – объектно-ориентированный ЯП Python

**Данная программа проходит апробацию с 10 по 11 класс с 2018 года. Закончит апробацию в 2020 году.**

Рекомендуемая учебная литература по данной программе:

1. А.А. Чернов «Программирование на языках высокого уровня». Волгоград: «Учитель», 2006,
2. «Основы программирования». С.М. Окулов и др., М.: Бином. Лаборатория Знаний, 2006,
3. М. Лутц «Изучаем Питон», Санкт-Петербург: Символ, 2011г.
4. К.Ю. Поляков, Е. А. Еремин. «Информатика», углубленный уровень, М.: Бином. Лаборатория Знаний, 2013.

## VII. Содержательные линии. Логика построения курса

Для формирования вышеуказанных компетентностей при изучении предмета "Основы программирования на Python" реализуются нижеследующие содержательные линии:

- **Алгоритмический (структурный) ЯП Python»**
- **Объектно-ориентированный ЯП Python**

Содержательная линия «**Алгоритмический (структурный) ЯП Python**» направлена на развитие логического и алгоритмического мышления, формирование умений составлять и реализовывать линейные алгоритмы, алгоритмы с ветвлением и повторением с использованием числовых, символьных и строковых величин, элементов одномерных массивов на языке программирования. Рекомендуемые темы в рамках данной содержательной линии:

- Алгоритм
- Условия, ветвления, циклы
- Среда программирования
- Математическая логика
- Компьютерная графика
- Языки программирования. Принципы построения. Блочные и формальные языки.
- Переменные. Типы данных.
- Функции.
- Алгоритмы сортировки, поиска.
- Модули программ.
- Рекурсия.
- Массивы. Алгоритмы обработки массивов.
- Абстракция и декомпозиция
- Ввод, вывод - файлы, консоль. Понятие кэширования. Двоичные, текстовые файлы
- Структуры данных. Списки.
- Строки, работа со строками. Форматирование строк.
- Матрицы.
- Сложные условия. Шаблоны
- Основы баз данных. Реляционные базы данных.
- ООП программирование. Базовые представление, основные принципы
- Основы робототехники

Содержательная линия «**Объектно-ориентированный ЯП Python**» направлена на развитие логического и алгоритмического мышления, формирование умений работать с графикой, графическим интерфейсом, библиотеками, виджетами, знания принципов ООП. Рекомендуемые темы в рамках данной содержательной линии:

- Концепция ООП. Объекты и классы
- Основные понятия объектно-ориентированного программирования, объектно-ориентированный анализ задачи.
- Программирование объектной модели. Практическое применение понятий, изученных ранее.
- Принципы ООП: инкапсуляция

- Принципы ООП: наследование и полиморфизм
- Основы графического интерфейса
- Создание приложения с использованием виджетов
- Создание графического интерфейса без использования программы-визуализатора
- Управление макетом графического интерфейса: блочный макет
- Управление макетом графического интерфейса: сеточный макет
- Рисование на форме
- Модель обработки данных в приложении с графическим интерфейсом
- Представление в приложении с графическим интерфейсом
- Создание собственного виджета.

*Данный курс содержит 35 видеоуроков, автора Дмитрия Тарасова, 34 теста в формате Word и в формате MyTestX, практические работы и 4 задания для мини-проектов.*

## **VIII. Основное содержание учебного курса для 10 класса, Образовательные результаты и оценивание процесса**

### ***1. Введение в язык программирования Python. Его история. Первый запуск рабочей среды***

Эта тема отвечает на вопрос «Почему Python?». Особенности этого языка программирования и его история. Инструкция по скачиванию и установке языка Python с сайта его разработчиков и запуску рабочей среды языка.

### ***2. Структура программы на языке Python. Операции и переменные. Типы данных***

Программы на языке Python. Составные структурные единицы программ, такие как модули, инструкции и выражения. Понятие переменной, операции и типизацией данных в языке Python.

### ***3. Ввод и вывод данных в программах на языке Python***

Цель любой компьютерной программы – это обработка данных. Для этого программа должна принимать данные на вход и возвращать результат их обработки пользователю. Функция ввода и вывода данных в программах на языке Python, а также различные варианты их использования. Пользовательский интерфейс программы и принципе его дружелюбности.

### ***4. Линейные алгоритмы обработки целочисленных данных***

Различные типы данных в языке Python. Целочисленный тип данных, инструкции и алгоритмы обработки таких данных. Первый тип структуры алгоритмов – линейные алгоритмы.

### ***5. Линейные алгоритмы обработки вещественных чисел***

Линейные алгоритмы обработки данных, а также с числовые типы данных. Числа с плавающей точкой в языке Python, инструменты для работы с ними, встроенные в рабочую среду, а также описанные во внешних модулях, таких как math и random.

### ***6. Разветвляющиеся алгоритмы. Логический тип данных. Инструкция ветвления***

Разветвляющиеся алгоритмы, инструментах для их реализации. Логический тип данных, основные логические операции и инструкция ветвления, которая позволяет выбрать один из двух наборов инструкций для исполнения в зависимости от ситуации.

### ***7. Разветвляющиеся алгоритмы. Сложные условия. Каскадные ветвления***

Инструкции ветвления могут быть вложенными и как это усложняет структуру алгоритма, а

также о каскадных ветвлениях, которые позволяют разделять код программы более чем на две ветви.

### **8. Циклические алгоритмы. Цикл с предусловием**

Помимо линейных и разветвляющихся алгоритмов, выделяется ещё третий основной тип алгоритмов – циклические. Циклическими называются алгоритмы, содержащие структуру повторения, первый способ реализации этой структуры – цикл с предусловием и как этот цикл программируется на языке Python.

### **9. Циклические алгоритмы. Цикл с постусловием. Инструкции управления циклом**

Цикл с постусловием. В языке Python нет отдельной инструкции для реализации именно этого цикла, но его можно реализовать с помощью инструкций управления циклом. Инструкции управления циклом, их программирование и необходимость использования.

### **10. Циклические алгоритмы. Цикл с параметром**

Циклом с параметром. Как работает такой цикл и с какими особенностями связано его программирование на языке Python. Функция генерации диапазона значений.

### **11. Сложные циклические алгоритмы. Вложенные циклы**

В программах могут использоваться вложенные циклы. Проблемы, связанные с программированием вложенных циклов, такие как усложнение структуры программы и сильное увеличение числа повторений, из-за которого программа может медленно работать.

### **12. Этапы решения задачи на компьютере. Последовательное конструирование алгоритма**

Этапы решения задачи на компьютере и их особенности. Этап создания алгоритма. При необходимости разработки сложного алгоритма можно использовать принцип последовательного конструирования алгоритма, принцип его работы.

### **13. Функции**

Запрограммировать решение отдельных подзадач можно в отдельных функциях, которые потом при необходимости вызываются в различных местах программы.

### **14. Применение функций при решении задач**

Польза функций при решении задач. Важные в программировании принципы, такие как модульность и повторное использование кода, для чего они нужны и в чём заключается их польза при разработке сложных программ.

### **15. Рекурсия**

Что такое рекурсия и в чём её сложность. Рекурсивные алгоритмы и их особенности: когда нужно использовать рекурсию в программировании, а когда лучше обойтись без этого.

### **16. Списки**

Структурные типы данных на примере списков. Определение понятия “списки”, их польза в программировании, а также их особенности. Способы создания ввода и вывода элементов списков в программах на языке Python.

### **17. Обработка списков. Линейный поиск. Операции над элементами списков**

Добавление и удаление элементов списков, а также линейный поиск элемента в списке.

### **18. Обработка списков. Двоичный поиск и сортировка**

Стандартные алгоритмы обработки списков. Сортировка, для чего она нужна, алгоритмы

сортировки пузырьком, рекурсивный алгоритм быстрой сортировки элементов списка. Как можно организовать поиск элемента в отсортированном списке.

### **19. Символьные строки**

Структурный тип данных – символьные строки. Как хранится текст в оперативной памяти компьютера, сходства и отличия между списками и символьными строками, основные операции обработки строк.

### **20. Матрицы**

Что такое матрицы и для чего они нужны, реализация матриц на языке Python и особенности их хранения и обработки, какие матрицы называются квадратными, свойства таких матриц, и применение их при решении задачи.

### **21. Обработка исключений**

В ходе исполнения компьютерных программ часто возникают ошибки. Они могут быть вызваны самыми разными причинами. Основные инструменты их предотвращения – обработка исключений, типы исключений, которые возникают чаще всего.

### **22. Работа с файлами**

При работе со структурными типами данных ввод данных с клавиатуры и их вывод на экран имеют целый ряд недостатков. Так, ввод данных с клавиатуры занимает много времени, а при выводе на экран полученные данные сложно использовать где-то ещё. Эти недостатки устраняет файловый ввод-вывод данных. Какие есть основные инструменты обработки файлов в языке Python, а также программирование файловых ввод-выводов при решении нескольких задач.

## **IX. Основное содержание учебного курса для 11 класса**

### **1. Концепция ООП. Объекты и классы**

Основные понятия объектно-ориентированного программирования, объектно-ориентированный анализ задачи.

**2. Программирование объектной модели.** Практическое применение понятий, изученных ранее.

### **3. Принципы ООП: инкапсуляция**

Помимо основных понятий, в объектно-ориентированном программировании есть ряд принципов, которые позволяют писать краткий и надёжный код. Инкапсуляция - позволяет скрывать внутреннее устройство объектов от внешней среды.

### **4. Принципы ООП: наследование и полиморфизм**

Ещё два принципа объектно-ориентированного программирования: наследование и полиморфизм. Эти принципы позволяют существенно сократить объектный код, не ограничивая при этом его функциональность. В каком случае один класс является наследником другого, как классы-наследники могут по-разному реализовывать методы своего предка и как это реализуется при программировании.

### **5. Основы графического интерфейса**

До этого все рассмотренные программы управлялись через текстовый интерфейс, однако большинство современных программ имеет графический пользовательский интерфейс. Особенности графического пользовательского интерфейса, принцип его работы, а также установка и запуск графической библиотеки PyQt5.

## **6. Создание приложения с использованием виджетов**

Как создать простое приложение с графическим интерфейсом пользователя. Графический интерфейс создан с помощью программы-визуализатора QtDesigner, входящей в состав графической библиотеки PyQt5.

## **7. Создание графического интерфейса без использования программы-визуализатора**

До этого все рассматриваемые графические интерфейсы создавались с помощью программы-визуализатора, однако у таких программа есть один существенный недостаток. Код, который генерирует такая программа, далеко не самый рациональный. Как создать графический интерфейс приложения с помощью библиотеки PyQt5 без использования программы-визуализатора.

## **8. Управление макетом графического интерфейса: блочный макет**

Как сделать так, чтобы элементы управления графическим интерфейсом адаптировались под изменённый размер окна программы. Для решения используется блочный макет окна программы. Блочный макет графического интерфейса для одной из программ, написанных ранее.

## **9. Управление макетом графического интерфейса: сеточный макет**

Ещё один тип макетов, применяющихся в графической библиотеке PyQt5, – сеточные – и принцип их работы. Политика размеров элементов управления, для чего она нужна и какой бывает.

## **10. Рисование на форме**

Для чего нужно рисование при создании приложений с графическим интерфейсом, какие классы описаны для рисования в графической библиотеке PyQt5 и как запрограммировать рисование.

## **11. Модель обработки данных в приложении с графическим интерфейсом**

Для чего в приложениях с графическим пользовательским интерфейсом принято выделять в отдельные модули модель обработки данных и представление, как соотносятся эти части между собой и что они содержат. Создание модели обработки данных для приложения «Калькулятор».

## **12. Представление в приложении с графическим интерфейсом**

Составные части приложения с графическим интерфейсом. Представление для приложения «Калькулятор» и связь модели обработки данных и представления. Что такое контроллер и за что он отвечает в приложении с графическим интерфейсом.

**13. Создание собственного виджета.** Но как быть, если необходимого виджета нет в графической библиотеке? Собственный виджет с помощью средств графической библиотеки PyQt5.

## **Х. Образовательные результаты и оценивание процесса. Основные стратегии оценивания достижений учащихся**

Система оценивания – основное средство измерения достижений и диагностики проблем обучения, осуществления обратной связи, оповещения учеников, учителей, родителей, государственных и общественных структур о состоянии, проблемах и достижениях образования в обществе.

Как известно, повышению качества обучения в значительной степени способствует правильная организация проверки, учета и контроля знаний учащихся и тесно связана с оценкой. Качество приобретаемых знаний, желание учащихся обучаться в дальнейшем зависит от объективности получаемой оценки.

Объектами оценивания в классе являются индивидуальные образовательные достижения и прогресс учащихся.

Для измерения образовательных достижений и прогресса учащихся применяются три вида оценивания: *диагностическое, формативное и суммативное.*

### **Диагностическое оценивание.**

Для оценки прогресса учитель в течение учебного года проводит сопоставление начального уровня сформированности компетентностей учащегося с достигнутыми результатами. Результаты диагностического оценивания регистрируются в виде описаний, которые обобщаются и служат основой для внесения коррективов и совершенствования процесса обучения путем постановки задач обучения для учителя и учебных задач для учащегося.

### **Формативное оценивание.**

Цели формативного оценивания – определение успешности и индивидуальных особенностей усвоения учащимися материала, а также выработка рекомендаций для достижения учащимся успеха. Учитель использует формативное оценивание для своевременной корректировки обучения, внесения изменений в планирование, а учащийся – для улучшения качества выполняемой им работы. Оценивается конкретная работа, выполненная учащимся, но не уровень его способностей.

При оценке промежуточных результатов обучения учитываются особенности учащихся (темпы выполнения работы, способы освоения темы и т.п.), фокусируется внимание на достижениях и прогрессе учащихся. Прогресс учащегося определяется как достижение определенных результатов, заложенных в целях обучения в рамках образовательных областей. Отметка в журнале регистрируется по необходимости, учитель фиксирует собственные наблюдения индивидуального прогресса учащихся.

### **Суммативное оценивание.**

Суммативная оценка учащихся служит для определения степени достижения учащимся результатов, планируемых для каждой ступени обучения, и складывается из текущего, промежуточного и итогового оценивания.

**Текущий контроль** используется после каждого урока для оценивания уровня усвоения материала классом.

**Промежуточный контроль** используется по итогам изучения отдельной темы (учебного модуля).

**Итоговый контроль** осуществляется по итогам полугодия, года, а также как итоговая аттестация при завершении предмета.

Основные требования к уровню знаний:

- ✓ при текущем контроле проверке подлежат лишь вопросы, затронутые на предыдущем занятии;
- ✓ при тематическом контроле подлежат проверке знания, зафиксированные необходимыми нормативными документами
- ✓ итоговый контроль осуществляется при переходе с одной ступени на другую и предполагает наличие необходимого минимума знаний для дальнейшего обучения.

Очень часто учителя использует отметки в качестве расправы с неугодными учениками. Такой подход не позволителен. Контроль должен рассматриваться как средство изучения уровня усвоения знаний. При низком усвоении учебного материала необходимо пересмотреть уровень преподавания, продумать изменение форм обучения и подходов к стилю обучения. Уже на начальном этапе изучения материала ученики четко должны представлять, к какому итогу, результату они должны подойти.

При четкой организации деятельности учителя и учеников, когда каждый из участников учебного процесса осознанно фиксирует свои результаты труда, другими словами осуществляет самоконтроль, тогда воспитывающая и обучающая роль оценки многократно возрастает. При этом учитель вовремя принимает необходимые меры для улучшения организации труда, а ученик начинает критически относиться к уровню собственного знания и выстраивает собственную траекторию самообразования.

В качестве традиционных методов проверки теоретических знаний можно использовать устный опрос, тестирование. Для оценивания практических навыков можно использовать практическую работу. В качестве нетрадиционных методов контроля можно использовать задачи на создание групповых проектов в командах. В качестве итогового контроля может быть использован (как групповой, так и индивидуальный) проект, где будут отражены как теоретические знания учащихся, так и уровень прикладных навыков работы с различными программными продуктами.

Устный опрос осуществляется на каждом уроке, когда не обязательно оценивать знания учащихся. Главным условием деятельности учителя является определение проблемных мест в усвоении учебного материала и фиксирование внимания учеников на сложных понятиях, явлениях, процессе.

Практическая работа включает в себя описание условия задачи без необходимых указаний, что делать, т.е. является формой контроля усвоения знаний. Следует отметить, что практическая работа зачастую связана с заданием на компьютере, например, может быть дано задание построения схемы, таблицы, написания программы и т.д.

Особенно остановимся на тестировании, как виде контроля. Грамотно составленные тесты могут быть не только формой контроля знаний, но и средством повторения и закрепления пройденного материала. Для использования тестов в качестве итогового контроля, необходимо регулярно тестировать учащихся в течение учебного года. Эффективным средством обучения является использование тестов в качестве описания конечных результатов деятельности.

На уроках информатики применяются два типа педагогического оценивания: критериальное и нормативное.

Критериальное оценивание – это оценивание по критериям, т. е. оценка складывается из составляющих (критериев), которые отражают уровень достижений учащихся. Критериальный подход к оцениванию учебных достижений учащихся на уроках информатики предполагает осуществление обратной связи, позволяющей всем участникам процесса (и учителям, и ученикам, и родителям) понимать уровень освоения изучаемого материала.

Примеры критериев оценивания на уроках информатики:

- умение учащегося логически обосновать решение поставленной задачи;
- умение ученика применить алгоритмический подход к выполнению задачи;
- умение учащегося выбрать оптимальный способ решения задачи;
- умение ученика сопроводить решение задачи необходимыми объяснениями;
- умение ученика охарактеризовать и проанализировать свое исполнение;
- умение ученика выполнять задачи в заданный срок;
- оригинальность ответа на вопрос или оригинальность решения задачи, которые свидетельствуют о высоком уровне владения информационными технологиями учащимся.

Критерии оценки продуктов деятельности учащихся разрабатываются учителем или совместно с учащимися к каждому виду работы и заданию.

Нормативное оценивание также является обязательным на уроках информатики - это подход к оценке и интерпретации измеряемых заданием показателей, отражающих особенности личности или поведения, путем сравнения индивидуальных результатов со статистическими значениями - нормой. Результаты обучения оцениваются по пятибалльной шкале, при необходимости дополняются устной характеристикой ответа.

***Ориентировочная модель системы оценивания индивидуальных образовательных достижений учащихся***

<b>№</b>	<b>Виды работ</b>	<b>Формы</b>
<b>1</b>	<b>Текущее оценивание</b>	
1.1.	Определяет учитель	Устный ответ, самостоятельная работа, домашние задания, презентация, практическая работа, компьютерный тренажер, компьютерное тестирование, игры
<b>2.</b>	<b>Промежуточное оценивание</b>	
2.1.	Письменные работы/ работа с источниками	Реферат, поиск дополнительной информации
2.2.	Устный ответ / презентация	Сообщение, доклад, презентация, вопросы-ответы, деловые игры, викторина
2.3.	Проект, исследовательская работа, специальные виды работ	Исследовательский отчет, описание результатов экспериментальных/лабораторных работ, тематические проекты, практические работы
2.4.	Портфолио (папка достижений), галерея	Демонстрационное, накопительное, творческое портфолио
<b>3.</b>	<b>Итоговое оценивание</b>	
3.1.	Четвертная, полугодовая, проверочная / контрольная работа	Компьютерный тест, итоговая контрольная работа по вариантам, итоговый проект (готовая написанная программа, интерактивная игра, мобильное приложение)

## **XI. Нормы отметок**

### **Для устных и письменных ответов:**

#### **- отметка «5» выставляется, если ученик:**

- изложил материал грамотным языком в определенной логической последовательности;
- правильно выполнил графическое изображение алгоритма и иные чертежи и графики, сопутствующие ответу;
- показал умение иллюстрировать теоретические положения конкретными примерами, применять их в новой ситуации при выполнении практического задания;
- продемонстрировал усвоение ранее изученных сопутствующих вопросов, сформированность и устойчивость используемых при ответе умений и навыков;
- отвечал самостоятельно без наводящих вопросов учителя.

#### **- отметка «4» выставляется, если:**

- ответ удовлетворяет основным требованиям, но при этом имеет один из недостатков:
  - в изложении допущены небольшие пробелы, не исказившие логического и информационного содержания ответа;
  - допущены один-два недочета при освещении основного содержания ответа, исправленные по замечанию учителя;
  - допущены ошибка или более двух недочетов при освещении второстепенных вопросов или в выкладках, легко исправленные по замечанию учителя.

#### **- отметка «3» выставляется, если:**

- неполно или непоследовательно раскрыто содержание материала, но показано общее понимание вопроса и продемонстрированы умения,
- имелись затруднения или допущены ошибки, исправленные после нескольких наводящих вопросов учителя;
- испытывает затруднения при ответе на видоизмененные вопросы;

#### **- отметка «2» выставляется, если:**

- обнаружено незнание или непонимание учеником большей или наиболее важной части учебного материала,
- допущены ошибки, которые не исправлены после нескольких наводящих вопросов учителя.

#### **- отметка «1» ставится, если:**

- обнаружено незнание или непонимание учеником всего учебного материала,
- допущены ошибки, которые не исправлены после повторного объяснения материала учителем.

### **Для самостоятельных практических заданий**

#### **- отметка «5» ставится, если:**

- учащийся самостоятельно выполнил все этапы решения задач на компьютере;
- работа выполнена полностью и получен верный ответ или иное требуемое представление результата работы;

#### **- отметка «4» ставится, если:**

- работа выполнена полностью, но при выполнении обнаружилось недостаточное владение навыками работы с компьютером в рамках поставленной задачи;

- правильно выполнена большая часть работы (свыше 75%);
- работа выполнена полностью, но использованы наименее оптимальные подходы к решению поставленной задачи.
- **отметка «3» ставится, если:**
  - работа выполнена не полностью (менее 75%), но учащийся владеет основными навыками для решения поставленной задачи.
- **отметка «2» ставится, если:**
  - допущены существенные ошибки, показавшие, что у учащегося не достаточно сформированы компетентность для решения поставленной задачи или значительная часть работы выполнена не самостоятельно.
- **отметка «1» ставится, если:**
  - допущены ошибки, показавшие, что у учащегося не сформирована компетентность для решения поставленной задачи или работа полностью выполнена не самостоятельно.

#### Для тестов:

- **отметка «5» ставится, если:**
  - учащийся дал более 90% правильных ответов.
- **отметка «4» ставится, если:**
  - учащийся дал от 75% до 89% правильных ответов;
- **отметка «3» ставится, если:**
  - учащийся дал от 60% до 74% правильных ответов;
- **отметка «2» ставится, если:**
  - учащийся дал от 20% до менее 59% правильных ответов.
- **отметка «1» ставится, если:**
  - учащийся дал менее 20% правильных ответов.

## ХII. Список использованной литературы:

1. «Государственный образовательный стандарт среднего общего образования». Постановление Правительства Кыргызской Республики, г. Бишкек, от 21 июля 2014 года №403;
2. Предметный стандарт «Информатика 5-9»
3. Болотов В.А., Сериков В.В. Компетентностная модель: от идеи к образовательной программе: – М. Педагогика, 2003. №10. – С.8-14.
4. Бородин М.Н., Цветкова М.С. Роль предмета Информатика и ИКТ в современном естественно-научном образовании. Базисный аспект. М: Издательство «БИНОМ. Лаборатория знаний» (БИНОМ), 2008, [www.LBZ.RU](http://www.LBZ.RU)
5. Закон Кыргызской Республики «Об Образовании» от 30 апреля 2003 г., Бишкек
6. Иванова Т.В. основные тенденции разработки требований в Государственных образовательных стандартов к уровню подготовки выпускников /Стандарты и мониторинг в образовании. – М., 2003, №5. –С. 3-13.
7. Информационные и коммуникационные технологии в образовании. Основная образовательная программа подготовки специалиста по специальностям: 050202.65 *Информатика*. «Мурманский государственный гуманитарный университет»(ФГБОУ ВПО «МГГУ»), 2013
8. Кириллова В.В. Использование метода проектов на уроках информатики: Выступление на РМО учителей информатики. - Горьковское, 2008
9. Концепция профильного обучения на старшей ступени общего образования КР // Кут билим, 23 март 2007, – Бишкек, 2009.
10. Концепция учебного предмета «Информатика», Министерство образования Республики Беларусь, 2009
11. Краевский В.В., Хуторской А.В. Предметное и общепредметное в образовательных стандартах. – М., Педагогика, 2003. №2
12. Куррикулум по предмету «Информационно-коммуникационные технологии» для 7-9 классов. – Бишкек, 2010
13. Программа Правительства по реализации Национальной стратегии устойчивого развития КР на период 2013-2017 годы;
14. Программа развития Кыргызской Республики на 2018-2022 гг. “Единство. Доверие. Созидание.”, 2018, п.8.1.5, 8.1.6.
15. Рамочный Национальный куррикулум среднего общего образования Кыргызской Республики. – Б., 2010. – 66 стр.
16. Рекомендации по оценке знаний, умений и навыков учащихся по курсу «Информатика и ИКТ», М - 2009. [https://docs.google.com/document/d/19vUF6-AeIohhLybBfZiivs\\_SCR8buTrhR\\_6quP3Mimw/edit](https://docs.google.com/document/d/19vUF6-AeIohhLybBfZiivs_SCR8buTrhR_6quP3Mimw/edit)
17. Сухих А.Я. Применение метода проектов на уроках информатики. - М.2013 <http://videouroki.net/filecom.php?fileid=98662927>
18. Формативдик жана суммативдик баалоо боюнча колдонмо. – Б., 2008.
19. Хуторской А.В. Ключевые компетенции как компонент личностно-ориентированной парадигмы образования [Текст] / А.В.Хуторской // Ученик в обновляющейся школе. - М.: ИОСО РАО, 2002. - С.135-157.