

ПРИНЯТА

Педагогическим советом ОАНО «Школа ЦПМ»
(протокол от 29 августа 2025 г. №123)

УТВЕРЖДЕНА

приказом директора ОАНО «Школа ЦПМ»
от 29 августа 2025 г. №207/8-ОД25

РАБОЧАЯ ПРОГРАММА

Курса внеурочной деятельности «Основы программирования»

10-11 класс

Составитель:

Батайкин Г.

Андросов Александр Александрович

sn=Андросов Александр

Александрович, o=ОАНО Школа

ЦПМ, ou=Директор,

email=a.androsov@school-cpm.ru,

c=RU

2025.08.29 09:41:13 +03'00'

Москва

2025 год

Пояснительная записка

1. Общая характеристика учебного курса

Курс «Основы программирования» в 10 классе является частью общей программы по программированию, освоение которой продолжается в 11 классе. Курс преследует своей целью практическое изучение программирования от базового и функционального языка Python, который применяется в большинстве сфер сегодня, до более сложных и узкоспециализированных в старших классах. Курс составлен так, чтобы даже учащийся, не имеющий никакого опыта программирования смог свободно освоить курс с самых азов, а учащиеся, имеющие опыт программирования, смогли вспомнить базу для успешного освоения курса в дальнейшем. Освоение Python – первое погружение в среду актуального программирования, курс дает базу знаний для будущих разработчиков в любой сфере, позволяя реализовывать собственные проекты в течение курса и по его завершении, а также подготовиться к реализации проектов и сложных задач в высших учебных заведениях.

Среди целей освоения курса выделены следующие:

Профориентационная – практическое освоение языка Python позволяет определить наличие интереса к сфере программирования

Развивающая – изучение нового языка программирования формирует системное мышление и позволяет решать нетривиальные задачи

Подготовительная – курс позволит свободно программировать на языке Python и даст необходимую базу для освоения других языков программирования, так как принципы формирования и логика схожи между собой

2. Место учебного курса в учебном плане

Данный курс реализуется в рамках специализации «Основы программирования» Универсального профиля Школы ЦПМ и предлагается для включения в учебный план в 10-11 классе. Рекомендуется освоение предыдущих ступеней данного курса в 9-10 классах, если же ученик впервые приступает к изучению дисциплины в 11 классе, рекомендуется самостоятельное ознакомление учащегося с предыдущими главами курса по запросу.

3. Планируемые результаты освоения учебного предмета

Студенты, прошедшие данный курс, сформируют более полное представление о том, в каких сферах можно использовать полученные знания в области программирования; поймут, как реализовать эти знания в собственных проектах, а также с большей уверенностью смогут выбрать подходящую специализацию при поступлении в высшее учебное заведение.

Личностные результаты:

1. Формирование критического мышления: Учащиеся учатся анализировать проблемы и оценивать различные подходы к их решению.
2. Развитие устойчивости при неудачах: При отладке программ учащиеся учатся воспринимать ошибки как часть учебного процесса и не сдаваться.
3. Увеличение интереса к технологиям: Учащиеся начинают осознавать важность программирования в современном мире, что может вызвать стремление развивать свои навыки дальше.
4. Формирование ответственности: Учащиеся учатся планировать выполнение проектов, устанавливать сроки и нести ответственность за результаты своей работы.
5. Сотрудничество и командная работа: При выполнении групповых проектов учащиеся развивают навыки взаимодействия и совместного решения задач.

Метапредметные результаты:

1. Навыки организации работы: Учащиеся учатся планировать и структурировать свои действия при выполнении программных заданий.
2. Критическое восприятие информации: Учащиеся учатся анализировать исходные данные и выбирать наиболее подходящие методы их обработки.
3. Развитие алгоритмического мышления: Учащиеся учатся разрабатывать и применять алгоритмы для решения задач в различных ситуациях.
4. Навыки самообучения: Учащиеся смогут искать необходимую информацию и изучать дополнительные ресурсы для углубления своих знаний.
5. Проблемное обучение: Учащиеся учатся ставить перед собой цели и находить способы их достижения, используя системный подход.

Предметные результаты:

1. Основы синтаксиса Python: Учащиеся овладеют основными конструкциями языка, такими как переменные, типы данных, операторы и функции.
2. Создание простых программ: Учащиеся смогут разрабатывать простые программы, используя навыки структурирования кода и базы алгоритмов.
3. Работа с библиотеками: Учащиеся научатся использовать стандартные библиотеки Python для решения различных задач.
4. Отладка кода: Учащиеся освоят основные методы отладки программ и технику поиска ошибок в коде.
5. Общие принципы объектно-ориентированного программирования: Учащиеся получат представление о принципах ООП и научатся применять их при написании собственных классов и объектов.

Полученные за время обучения знания сформируют необходимую базу для дальнейшего самостоятельного обучения, а также упростят обучения на первых курсах обучения в технических вузах на программах, связанных с компьютерными науками.

Содержание учебного предмета

Блок 1. ООП

Данный блок является продолжением изучения ООП, начатое годом ранее. Студенты познакомятся с таким понятием как наследование, которое потребуется при изучении фреймворка PyQt. Научатся обрабатывать исключения приводящие к завершению работы программы и получают первый опыт формирования предсказуемого поведения приложения. Также студенты познакомятся с магическими методами классов, которые позволят написать собственные итерируемые структуры данных, а также перегружать операторы.

Блок 2. Введение в разработку графического интерфейса

В блоке "Введение в разработку графического интерфейса" студенты освоят основы создания пользовательских интерфейсов с помощью PyQt. Они научатся создавать простейшие оконные приложения, разберутся в структуре и функционале виджетов, а также изучат методы организации многостраничных интерфейсов. Особое внимание будет уделено интеграции графических интерфейсов с базами данных, что позволит студентам создавать приложения, взаимодействующие с информацией в реальном времени. В практической части студенты создадут собственное приложение с графическим интерфейсом.

Блок 3. Проектирование и разработка реляционных баз данных

В блоке "Проектирование и разработка реляционных баз данных" студенты не только познакомятся с основами работы с реляционными базами данных, но и научатся их проектировать под конкретную бизнес-задачу. Научатся создавать и управлять базами данных. В рамках занятий студенты изучат, как извлекать, обновлять и удалять данные, а также поймут, как структурировать информацию для обеспечения высокой производительности запросов. Практические задания будут включать создание собственной базы данных и выполнение запросов с использованием Python.

Блок 4. Создание функционально приложения

Данный блок направлен на умение комбинирования полученных знаний и технологий. Студенты отойдут от концепции «один проект – одна библиотека» и получат первый опыт организации совместной работы нескольких инструментов для достижения поставленной задачи. В рамках блока каждый ученик разработает свое desktop-приложения, которое выступит связью между пользователем и спроектированной студентом базой данных. Сформированная таким образом информационная система не будет иметь претензии на статус готового бизнес-решения, но позволит сформировать понимание, как происходит передача и редактирование информации внутри базы данных.

Блок 5. Защита итоговых проектов

В заключительном блоке программы студенты пройдут процедуру защиты разработанных в течение года desktop-приложений. В рамках защит каждому ученику предстоит подготовить доклад-презентацию в которой ему предстоит описать процесс разработки, продемонстрировать работоспособность приложения и ответить на вопросы преподавателя, будучи в роли спикера перед одноклассниками.

Тематическое планирование учебного предмета

№ п/п	Тема	ЭОР	Количество ак. часов
1	ООП		10
1.1	Вспоминаем прошлый год		2
1.2	Наследование	https://metanit.com/python/tutorial/7.3.php	2
1.3	Магические методы	https://metanit.com/python/tutorial/7.7.php	2
1.4	Простейшие структуры данных		2
1.5	Исключения	https://metanit.com/python/tutorial/2.11.php	2
2	Введение в разработку графического интерфейса		20
2.1	Установка ПО и Изучение среды qt-designer	https://itproger.com/course/python-pyqt5	2
2.2	Создание кнопок и лейблов	https://doc.qt.io/qtforpython-6/tutorials/basic/tutorial/clickablebutton.html	2
2.3	Изучение Лэйаутов	https://doc.qt.io/qt-6/layout.html	2
2.4	Изучение виджетов (radio button, check box, combo box, date edit)	https://doc.qt.io/qt-6/qtwidgets-index.html	6
2.5	Таб-виджет		2
2.6	Изучение валидаторов	https://doc.qt.io/qtforpython-6/PySide6/QtGui/QValidator.html	2
2.7	Способы кастомизации приложения в qt-designer		2
2.8	Работа с изображениями в qt-designer		2
3	Проектирование и разработка реляционных баз данных		16
3.1	Концептуальное проектирование	https://www.decosystems.ru/kontseptualnoe-proektirovanie-bazy-dannykh/	8
3.2	Установка ПО и создание однотобличной БД	https://pypi.org/project/SQLAlchemy/	2
3.3	Создание консольного интерфейса для БД	https://docs.sqlalchemy.org/en/20/orm/quickstart.html	2
3.4	Создание связанных таблиц		4
4	Создание функционально приложения		16
4.1	Визуализация табличных данных	https://doc.qt.io/qtforpython-6/tutorials/basic/tutorial/tablewidget.html	2
4.2	Создание простейшего интерфейса для однотобличной БД	https://habr.com/ru/articles/336478/	6
4.3	Интеграция Таб-виджет для визуализации нескольких таблиц		4
4.4	Создание интерфейса для связанных таблиц		4
5	Защита проектов		6

Методика оценки успеваемости студентов

Оценка годовой успеваемости студентов основывается на среднем бале по итогу двух семестров обучения. Каждый из двух семестров имеет свою методику оценивания.

Первое полугодие:

На протяжении всего года студенты будут получать домашние работы, которые оцениваются по пятибалльной шкале. Каждые 3 недели первого полугодия будет происходить фиксирование достигнуты результатов (среднего балла за домашние работы и работы на занятии в рамках прошедших 3 недель). Таким образом ученики получают 5 оценок, среднее значение от которых станет их оценкой за текущее полугодие.

Второе полугодие:

Для оценки второго полугодия также, как и в первом студенты наберут по 5 оценок. Первые три будут получены по итогам выполнения домашних работ и работы на занятиях и будут фиксироваться каждые 6 недель. Последняя оценка зарабатывается по итогу защиты годового проекта. Для подсчета конченной оценки за полугодие будет использована следующая формула:

$$M = \frac{X_1 + X_2 + X_3 + Y}{4}$$

Где,

- X_i – оценки за домашнюю и аудиторную работу на протяжении первых 12 недель;
- Y – оценка за итоговый проект.

За особые достижения на занятиях и при выполнении домашних работ, студенты будут получать «купоны», которые позволят им поднять оценку за отдельно взятую домашнюю работу, которая выполнена без нареканий и грубых ошибок, но дедлайн для которой уже вышел. Один купон может поднять оценку не более чем на 1 балл. Также студенты могут ими обмениваться и при желании выручать товарищей. Единоmomentно на счету одного ученика не может быть более 5 купонов.