

Министерство просвещения РФ  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Уральский государственный педагогический университет»

Институт математики, физики и информатики  
Кафедра информатики, информационных технологий  
и методики обучения информатики

*На правах рукописи*

ГЛАДКИХ Татьяна Григорьевна

**ФОРМИРОВАНИЕ УМЕНИЯ ВЫПОЛНЯТЬ  
ОТЛАДКУ ПРОГРАММЫ ФОРМИРОВАНИЯ  
НАВЫКА ПРОВОДИТЬ ОТЛАДКУ ПРОГРАММЫ  
НА ОСНОВЕ ФОРМИРОВАНИЯ ВЫДЕЛЕННЫХ  
КОМПОНЕНТОВ**

Диссертация на соискание степени  
магистра образования

Направление «44.04.01 – Педагогическое образование»

Магистерская программа «Цифровые технологии  
в работе педагога»

Научный руководитель:  
кандидат педагогических  
наук,  
доцент А.И. Газейкина

Екатеринбург 2023

## Оглавление

<b>ВВЕДЕНИЕ.....</b>	<b>3</b>
<b>ГЛАВА 1. ТЕОРЕТИЧЕСКИЕ ОСНОВАНИЯ ОРГАНИЗАЦИИ УЧЕБНОГО ПРОЦЕССА ДЛЯ ФОРМИРОВАНИЯ УМЕНИЯ ПРОВОДИТЬ ОТЛАДКУ ПРОГРАММЫ.....</b>	<b>8</b>
1.1. ПЕДАГОГИЧЕСКИЕ ОСНОВЫ ОБУЧЕНИЯ ПРОГРАММИРОВАНИЮ В СРЕДНЕЙ ШКОЛЕ.....	8
1.2. СРЕДСТВА ФОРМИРОВАНИЯ УМЕНИЯ ОТЛАДКИ ПРОГРАММНОГО КОДА....	17
1.3. ПРОЕКТИРОВАНИЕ МЕТОДИКИ ФОРМИРОВАНИЯ У ОБУЧАЮЩИХСЯ УМЕНИЯ ПРОВЕДЕНИЯ ОТЛАДКИ ПРОГРАММНОГО КОДА.....	20
ВЫВОДЫ ПО МАТЕРИАЛАМ ГЛАВЫ 1.....	25
<b>ГЛАВА 2. МЕТОДИКА ФОРМИРОВАНИЯ У ОБУЧАЮЩИХСЯ УМЕНИЯ ВЫПОЛНЯТЬ ОТЛАДКУ ПРОГРАММЫ.....</b>	<b>26</b>
2.1. КОМПОНЕНТЫ МЕТОДИКИ ФОРМИРОВАНИЯ УМЕНИЯ ВЫПОЛНЯТЬ ОТЛАДКУ ПРОГРАММЫ.....	26
2.2. УЧЕБНЫЕ ЗАДАНИЯ ДЛЯ ФОРМИРОВАНИЯ УМЕНИЯ ВЫПОЛНЯТЬ ОТЛАДКУ ПРОГРАММЫ.....	33
2.3. ОРГАНИЗАЦИЯ ОПЫТНО-ПОИСКОВОЙ РАБОТЫ И ЕЕ РЕЗУЛЬТАТЫ.....	37
<b>ВЫВОДЫ.....</b>	<b>39</b>
<b>ИСТОЧНИКИ ИНФОРМАЦИИ.....</b>	<b>41</b>
<b>ПРИЛОЖЕНИЯ.....</b>	<b>42</b>

## Введение

Общество ожидает от системы образования формирования интеллектуальной и технологической платформы для быстрого перехода в принципиально новое качественное состояние. Вместе с тем в начале XXI века в России стало явно проявляться несоответствие результатов деятельности отечественной системы образования требованиям, предъявляемым к ней. Это несоответствие вызвано отставанием содержания и технологий обучения от запросов современного общества и экономики страны.

Среднее образование направлено на формирование разносторонне развитой личности. Изучение программирования в средней школе позволяет обучающимся научиться осознавать и строить алгоритмические структуры, анализировать проблемы с разных точек зрения и искать пути их решения.

При изучении программирования учащиеся основной школы часто испытывают трудности. Умение построения алгоритмов и их представление их на языке программирования является для них принципиально новым. Помимо этого очень мало внимания уделяется отладке программного кода. Обучающиеся не видят как синтаксических, так и семантических ошибок и не умеют их искать. В области педагогики этой теме уделяется очень мало внимания. Проведя аналитический поиск научных работ по данной тематике, нам не удалось найти методики поиска и исправления программного кода для учащихся средней школы. Что удивительно, учитывая количество информации по данной теме в книжных магазинах.

ФГОС ОСО предусматривает овладения учащимися не только умением написания на одном из языков высокого уровня, но и умением отладки программного кода.

Проведенный анализ позволяет выделить ряд противоречий:

- на научно-педагогическом уровне – между дидактическими возможностями процесса отладки программы для развития критического

мышления школьников и недостаточной развитостью теоретических оснований обучения умения отладки программного кода;

- *на научно-методическом уровне* – между необходимостью овладения обучающимися умениями тестирования и отладки программного кода, что является важной частью освоения программирования и недостаточной разработанностью методики обучения тестированию программ и исправления найденных ошибок.

Необходимость разрешения перечисленных противоречий обуславливает актуальность данного исследования, а также его **проблему**: каким образом построить учебный процесс и какие учебные задания в него включить для освоения учащимися данных умений. В рамках указанной проблемы нами определена **тема исследования**: «Формирование умения выполнять отладку программы в процессе обучения программированию учащихся основной школы»

**Объект исследования**: процесс обучение программированию учащихся основной школы в учреждении дополнительного образования

**Предмет исследования**: формирование умения отладки программного кода у обучающихся основной школы на базе учреждения дополнительного образования

**Цель исследования**: научно обосновать и разработать методику формирования умения отладки программы в процессе обучения программированию у учащихся основной школы

При достижении поставленной цели мы руководствовались следующей **гипотезой**: умение осуществлять тестирование и отладку программы будет сформировано если:

- *при организации урочной и внеурочной деятельности по программированию будет сформировано умение анализировать программный код с целью нахождения синтаксических и семантических ошибок*

- будут поэтапно формироваться умения определять вид ошибки и способы ее исправления
- в соответствии с выделенными этапами будут использоваться специальные учебные задания: практическая работа, мини-исследование

На основании цели исследования и рабочей гипотезы были поставлены (сформулированы) следующие **задачи исследования:**

1) На основе анализа научно-методической литературы уточнить терминологический аппарат исследования (содержание понятия «отладка программного кода»), выделить элементы умения обучающегося проводить отладку программы.

2) Проанализировать дидактические возможности формирования умения проводить отладку программы у учащихся средней школы

3) Разработать педагогическую модель методики формирования умения проводить отладку программы в процессе изучения программирования

4) Разработать комплекс учебных заданий как основу методики формирования умения анализировать программу с целью поиска ошибок разного происхождения, исправлять их и проводить тестирование программного кода.

5) Осуществить опытно-поисковую работу по проверке результативности применения разработанной методики формирования умения проводить отладку программы в процессе обучения программирования.

**Теоретико-методологическую основу** исследования составили работы:

- методика обучения и содержание школьного курса информатики (Л.Л. Босова, А.Г. Гейн, И.Г. Семакин, Н.Д. Угринович, М. П. Лапчик);
- исследования в области тестирования и отладки программ (К.А. Кулаков, М. Плаксин);

**Методы исследования:** изучение научно-методической и специальной литературы по проблеме исследования; анализ государственных образовательных стандартов, программ, учебных пособий и методических

материалов; изучение и обобщение опыта учителей; педагогическое проектирование и моделирование; тестирование; анкетирование; статистическая обработка экспериментальных результатов.

**Методы исследования:** изучение научно-методической и специальной литературы по проблеме исследования; анализ государственных образовательных стандартов, программ, учебных пособий и методических материалов; изучение и обобщение опыта учителей; педагогическое проектирование и моделирование; тестирование; анкетирование; статистическая обработка экспериментальных результатов.

**Обоснованность и достоверность результатов исследования** и сделанных на их основе выводов обеспечивается фундаментальностью теоретических оснований работы, соответствием практических построений положениям и требованиям методики обучения информатике; логической непротиворечивостью всех разделов работы; использованием взаимодополняющих методов педагогического исследования; воспроизводимостью полученных результатов опытно-поисковой работы и подтверждением в ходе ее исходной гипотезы исследования.

**Научная новизна исследования** заключается в следующем:

1. В отличие от работ М. А Плаксина, А. Д. Хомоненко, Ю.А Волкова, исследование рассматривает процесс отладки программы на уровне начального обучения программированию в средней школе с опорой на существующую методику преподавания раздела программирования и особенностей рассматриваемого возраста

2. Разработана методика, применение которой обеспечивает формирование умения осуществлять отладку программного кода в процессе изучения раздела программирования на уроках информатики в средней школе.

3. Опытным путем подтверждена результативность применения предложенной методики.

**Теоретическая значимость** исследования заключается в следующем:

1. Уточнено понятие отладки программного кода.
2. Предложена педагогическая модель формирования навыка проводить отладку программы на основе формирования выделенных компонентов.
3. Выделены и обоснованы принципы поиска и исправления ошибок в программном коде.
4. Обоснованы критерии результативности применения методики формирования навыка проводить отладку программы на основе формирования выделенных компонентов.

**Практическая значимость** исследования состоит в том, что теоретические результаты доведены до уровня практического применения. Разработаны:

- совокупность специальных учебных заданий;
- методические рекомендации для учителей по применению методики формирования умения осуществлять отладку программного кода.

**Апробация и внедрение** основных идей и результатов исследования осуществлялась на базе детского технопарка «Кванториум» Свердловской детской железной дороги. В исследовании принимали участие учащиеся первого года обучения по направлению «Информационные технологии».

**Структура и объем диссертации.** Диссертация изложена на страницах, состоит из введения, двух глав, заключения, библиографического списка, включающего источников, приложений.

# **Глава 1. Теоретические основания организации учебного процесса для формирования умения проводить отладку программы**

## **1.1. Педагогические основы обучения программированию в средней школе**

Современное общество невозможно представить без деятельности программиста. Его работа зачастую не видна на первый взгляд, но не одно современное предприятие не обходится без него.

Согласно Федеральному образовательному стандарту среднего общего образования обучающиеся после выпуска из школы должны иметь сформированы основы логического и алгоритмического мышления и понимание необходимости формального описания алгоритмов. Так же делается упор на освоения умения составлять алгоритмы, умения их анализировать и проводить отладку написанного ими кода [8].

Курс информатики является обязательным для всех обучающихся и вне зависимости от уровня его освоения раздел программирования должен быть усвоен.

На основе анализа УМК различных авторов( И.Г. Семакина, Л.Л. Босовой)[12, 13] как базового, так и профильного профиля изучения информатики можно отметить, что основной упор при изучении программирования в школе идет на понимание и составление различных алгоритмических структур и типов данных и такая важная часть обучения как анализ программы с целью обнаружения и исправления ошибок остается без должного внимания, когда как именно данное умение и формирует понимание работы алгоритма и его правильной интерпретации на язык программирования высокого уровня.

Таким образом, можно сделать вывод о значимости и необходимости умения проводить тестирование и отладку программы для успешного освоения раздела программирования.

В научно-методической и технической литературе встречаются разные интерпретации понятий «тестирование» и «отладка».

М. П. Лапчик под отладкой программы понимает выполнение программы в отладочном режиме с остановкой после каждой команды [1].

Согласно С. В. Соловьеву отладка является процессом, позволяющим получить программу, функционирующую с требуемыми характеристиками в заданной области изменения входных [2].

Большая Российская энциклопедия определяет отладку программы, как процесс поиска, локализации и исправления ошибок в компьютерной программе, осуществляемую на всех этапах разработки программного обеспечения [3].

В научно-методической литературы рассматриваемое понятие часто упускается из вида. В связи с этим были проанализированы и популярные интернет-источники с целью уточнения исследуемого термина.

Отладка — это процесс локализации и исправления ошибок, обнаруженных при тестировании программного обеспечения [4].

Отладка – это процесс локализации и исправления ошибок, обнаруженных при тестировании программного обеспечения. Локализацией называют процесс определения оператора программы, выполнение которого вызвало нарушение нормального вычислительного процесса [5].

По итогу анализа литературных источников, определяющих понятие отладки программы, можно сделать вывод что все они делятся на два основных типа – включающие в процесс отладки процесс тестирования программы и те, которые выделяют тестирование как самостоятельный процесс, проводимый до этапа отладки.

Процесс тестирования в свою очередь так же является сложным процессом в рамках разработки программного обеспечения и требует комплексного подхода.

Тестирование программного обеспечения (Software Testing) — проверка соответствия реальных и ожидаемых результатов поведения программы, проводимая на конечном наборе тестов, выбранном определённым образом [6].

Тестирование (softwaretesting) — деятельность, выполняемая для оценки и улучшения качества программного обеспечения. Эта деятельность, в общем случае, базируется на обнаружении дефектов и проблем в программных системах. Тестирование программных систем состоит из динамической верификации поведения программ на конечном (ограниченном) наборе тестов (setoftestcases), выбранных соответствующим образом из обычно выполняемых действий прикладной области и обеспечивающих проверку соответствия ожидаемому поведению системы [7].

По результатам проведенного исследования определений «тестирование» и «отладки» программы следует, что в большинстве случаев невозможно провести качественную отладку программного кода без проведенного предварительно тестирования программы. Разбиение данных процессов на два самостоятельных этапа разработки программного обеспечения имеет смысл, если мы имеем дело с большими программами и многопоточными процессами, когда как в рамках изучения раздела программирования в среднем образовании изучаются лишь основы алгоритмизации и их интерпретация на языках высокого уровня. То есть, разрабатываемые программы являются небольшими и процесс тестирования значительно упрощается.

Исходя из этого в рамках данной работы понятие отладки будет рассматриваться как комплексный процесс, направленный на тестирование программы с целью обнаружения ошибок, их классифицирования на основе природы происхождения, локализации в программном коде с последующим устранением.

Процесс отладки программы является одним из ключевых этапов в разработке программного средства. От него зависит качество итогового продукта. Таким образом, представляется необходимым выделить алгоритм проведения отладки программы.

В большинстве литературных источников алгоритм отладки идентичен и включает в себя следующие этапы [9]:

- 1) проведение действий по выявлению ошибок (тестирование);
- 2) классифицирование природы возникновения найденных ошибок;
- 3) диагностика и локализация ошибок;
- 4) внесение исправлений в программу с целью устранения ошибок.

Для того чтобы сформировать умение отладки программ у обучающихся необходимо сформировать умения необходимые для проведения действий на каждом выделенном шаге алгоритма. В следствие чего, необходимо рассмотреть и проанализировать каждый этап подробнее с целью выявления универсальных учебных действий, лежащих в их основе.

#### 1. Проведение действий по выявлению ошибок.

Этот этап в данной работе, исходя из специфики методики обучения программирования на уровне среднего общего образования, является упрощенной версией тестирования программных средств. Он основывается на принципах тестирования для прикладного программного обеспечения, но не обязан содержать техническую документацию каждого теста, так как направлен на поиск примитивных ошибок, зачастую допущенных по невнимательности или низкого уровня математических знаний, необходимых для решения задачи.

Основная задача тестирования – проверка разработанного приложения на всем поле значений, как допустимых, так и предполагаемо невозможных. Тестирование является самым лучшим способом нахождения неточностей и ошибок в ходе. Перед проведением всех тестов необходимо разработать список входных значений, на котором будет проводится тестирование, и

список выходных данных, которые должны получиться в результате эталонного варианта работы программы [10].

В тестировании прикладных программ выделяют два основных вида: ручное и автоматическое.

При ручном тестировании вся работа ложится на плечи тестировщика без использования любых инструментов автоматизации. Цель данной работы – поиск ошибок в приложении. Является одним из наиболее простых способов выявления неработающих блоков кода. Любое тестирование программного средства начинается с ручного тестирования для поиска тривиальных ошибок, выявление которых не требует больших вычислений и глубоких знаний принципов работы тестируемой программы. Один из фундаментальных принципов тестирования гласит, что полный перевод тестирования в автоматический формат невозможен. Из чего следует, что при тестировании любого разработанного кода невозможно избежать ручного поиска ошибок.

Автоматическое тестирование же в свою очередь основано на использовании специализированных программных средств. Программное обеспечение такого рода способно вводить тестовые данные в тестовую среду, сравнивать ожидаемые и фактические результаты и создавать подробные отчеты о тестах. Данный тип тестирования требует больших финансовых затрат и других ресурсов. Он эффективен в случаях последовательного цикла разработки, когда возникает потребность проводить проверку одного и того же набора тестов. После такого рода тестирования не требуется вмешательство человека. Автоматизация тестирования преследует цель уменьшения количество тестовых примеров, которые нужно запускать вручную.

Исходя из вышесказанного в рамках данной работы мы будем рассматривать только ручное тестирование.

Тестирование программы является деструктивным процессом, то есть основная задача человека реализующего проверку сломать программу и правильно зафиксировать в следствии каких действий она появилась. Данный вид деятельности является диаметрально противоположным процессу разработки кода, из-за чего зачастую тестирование проводят люди, которые не занимались созданием программы в целях повышения эффективности проверки. В рамках же учебного процесса обе задачи, как разработки и тестирования необходимо выполнять одним и тем же лицам для повышения понимания допущенных неточностей и ошибок в программном коде и, следовательно, избегания их в будущем. Данный этап в процессе обучения программированию позволяет обучающимся посмотреть на разработанный алгоритм под другим углом и способствует развитию умения анализа информации.

Для проведения полного цикла тестирования программы обучающимся необходимо определить область допустимых значений и проверить работу программы на нем. На углубленном уровне изучения программирования обучающимся необходимо также выйти за пределы области допустимых значений и удостовериться, что программа не завершится аварийно.

## 2. классифицирование природы возникновения найденных ошибок

После возникновения ошибки на этапе тестирования или компиляции программы необходимо правильно классифицировать природу ее появления. Неправильное определение типа ошибки может привести к появлению новых ошибок и изменению правильно работающих блоков кода, что лишь отдалает от завершения разработки программы.

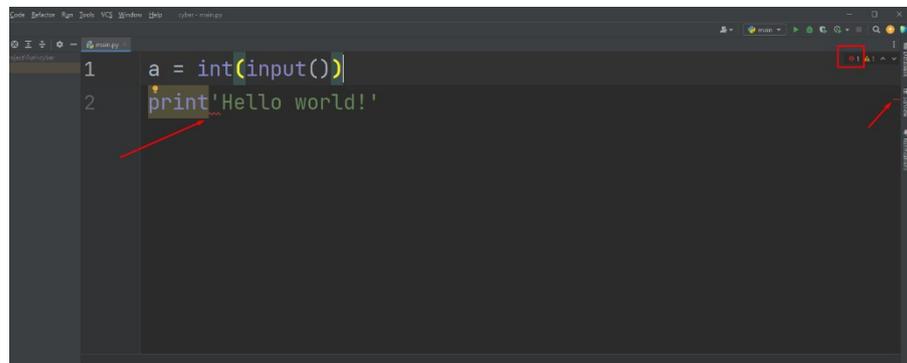
Это демонстрирует актуальность изучения типов ошибок и принципов их правильного их устранения.

Многие авторы выделяют классификации ошибок на 2 основных типа [11]:

- синтаксические;

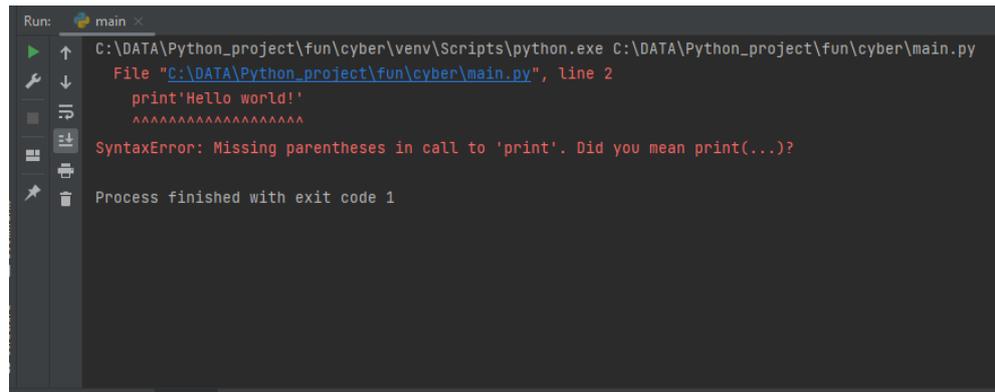
- семантические.

Синтаксические ошибки являются самыми легкими для обнаружения, так как они связаны с неправильным использованием правил написания определенного языка программирования высокого уровня. Все современные среды программирования включают в себя средства проверки синтаксиса языка, на котором происходит разработка. Ошибки данного типа видно еще до процесса компиляции. В зависимости от среды программирования синтаксические ошибки могут обозначаться по-разному, но они всегда каким-либо образом обозначают строку с ошибкой и подчеркивают красным цветом место с неправильно написанной командой (Рис. 1).



**Рис. 1 Поиск синтаксических ошибок в тексте**

В случаях, когда ошибка не была исправлена до процесса компилирования, программа завершится аварийно с типом допущенной синтаксической ошибки и кратким описанием ее природы и номером строки, в которой произошла ошибка. Зачастую при запуске программы с неисправленной ошибкой данного типа среда разработки сама предлагает вариант ее исправления (Рис. 2).



```

Run: main x
C:\DATA\Python_project\fun\cyber\venv\Scripts\python.exe C:\DATA\Python_project\fun\cyber\main.py
File "C:\DATA\Python_project\fun\cyber\main.py", line 2
print'Hello world!'
^^^^^^^^^^^^^^^^^^^^
SyntaxError: Missing parentheses in call to 'print'. Did you mean print(...)?

Process finished with exit code 1

```

**Рис. 2** Указания местоположения синтаксической ошибки на этапе компиляции

Отладка ошибок синтаксического типа является самой простой и зачастую не занимает много времени, но именно ошибки такого рода самые частые при первичном изучении программирования у обучающихся и с их помощью необходимо начинать обучать методам правильной отладки кода.

Семантические ошибки менее явные для нахождения до запуска программы. Они возникают на этапе компиляции программы и возникают из-за неправильного использования типов данных или структур программирования, обращения к несуществующему файлу или неправильной логики в алгоритме программы.

Существует два варианта завершения работы программы с содержащейся в ней семантической ошибкой: программа завершается аварийно с указанием конкретного типа ошибки или программа завершается без ошибок, но с неправильным результатом вычисления.

При первом варианте поиск и устранение возникшей ошибки очень похож на исправление синтаксических ошибок: в консоль среды разработки выводится конкретный тип возникшего исключения с указанием конкретной строки, в которой возникла ошибка. В данном случае сложность зачастую состоит именно в правильном определении неправильно использованной команды или указания пути к файлу.

Ошибки, которые не приводят к аварийной остановке программы можно найти только с помощью тестирования программы на заранее подготовленном наборе входных данных. Такие ошибки являются самыми

опасными для обучающихся на начальном этапе изучения программирования, так как они никак не выделяются средой разработки и требуют точного понимания работы каждого блока алгоритма и дальнейшего детального анализа программы построчно.

### 3. Диагностика и локализация ошибок.

Проведение диагностики разработанного кода и определения точно места возникновения ошибок необходимо в случае семантических ошибок, которые не приводят к аварийному завершению программы.

Для определения точного местонахождения неправильной работы алгоритма при изучении программирования на уровне среднего образования подходят следующие два метода: аварийная печать по пути работы программы и пошаговое выполнение программного кода.

Зачастую выбор метода локализации зависит непосредственно от разработчика программы, так как оба метода не могут дать однозначного представления и причины возникновения логической ошибки.

Метод аварийной печати представляет собой использования команды для вывода информации в консоль среды разработки для отображения каких-либо данных, необходимых для определения правильности работы программы и последующего их анализа исходя из требований решаемой задачи.

Метод пошагового выполнения программы состоит в создании точки остановки программы и включения специального режима выполнения программы, после запуска которого программа выполняется в стандартном режиме до точки остановки и последующего ручного выполнения. При данном методе разработчик видит наглядно изменение каждой переменной в программе на каждом шаге работы алгоритма, что позволяет более детально анализировать алгоритм и сопоставлять получаемые результаты с эталонными.

### 4. Внесение исправлений в программу с целью устранения ошибок.

После успешной локализации возникшей ошибки, разработчику необходимо перепроверить интерпретацию разработанного алгоритма на язык программирования на случай возможных опечаток, так как зачастую именно вследствие этого возникают ошибки на начальном этапе обучения программированию. Если алгоритм был интерпретирован верно, но ошибка не была исправлена, то возникает необходимость проанализировать составленный алгоритм с целью поиска в нем логических ошибок и переработать блок алгоритма, в котором возникает ошибка.

Исходя из всего вышесказанного, можно сделать вывод о необходимости и актуальности формирования умения проведения тестирования и отладки программного кода на ранних стадиях обучения программированию с целью успешного освоения рассматриваемого раздела информатики.

## **1.2. Средства формирования умения отладки программного кода**

На формирование и развитие умений обучающихся в первую очередь влияют задания, которые они решают в процессе обучения на уроках. Существует огромное количество различных типов заданий, каждый из которых предполагает акцентирование внимание обучающегося на одном или двух видах деятельности. Так же на выбор заданий для учебного процесса влияет специфика учебного предмета, а зачастую и особенности изучаемого раздела в рамках одной области знаний.

Учебное задание является общим определением и включает в себе задачу, данную в рамках определенной учебной ситуации. Перед выбором конкретных типов заданий необходимо учесть условия, в которых предполагается решения задания, то есть конкретную учебную ситуацию.

Выделяют три основных типа учебных задания по типу взаимодействия с преподавателем:

- предполагающие пассивную роль учащихся и активную подачи учебного материала преподавателем;
- предполагающие активную самостоятельную работу учащихся в рамках небольших групп или самостоятельно без прямого участия преподавателя;
- предполагающие активную самостоятельную работу обучающихся в рамках небольших групп или самостоятельно в комплексе с активным взаимодействием с преподавателем.

Для освоения умения проведения тестирования и последующего исправления появившихся ошибок наиболее подходящим типом учебных заданий являются те, где основной упор идет на активную работу обучающихся без постоянной преподавательской поддержки, так как такого рода задания способствуют концентрации внимания обучающихся на вникание в суть задания и саморегуляции своих действий. Активное взаимодействие с преподавателем возможно при выполнении первых заданий, где он демонстрирует способы их решения, в последующем такие действия могут привести к нежелательному результату, так как поиск ошибок и дальнейшее продумывание способов их избегания когнитивный процесс, на прямую связанный с индивидуальными особенностями обучающихся.

Так же можно разделить учебные задания по типу их решения:

- типовые задачи, которые имеют заранее продуманный алгоритм решения, заточенный под конкретную тему, и решаются по аналогии с эталонным примером;
- аналитические, которые требуют переосмысления известных алгоритмов в соответствии с анализом конкретных условий. В подобных задачах условия зачастую могут избыточными или недостаточными;

- творческие, которые требуют отвлечения от ранее усвоенных способов решения, нахождения новых способов решения и зачастую содержат в себе поисковую и исследовательскую составляющую;
- самообучающие. Особый тип заданий, которые при должной готовности обучающихся к самообучению и поисково-исследовательской деятельности при снабжении необходимой инструкцией могут использоваться обучающимися без соответствующей помощи и содействия учителя и влияет на качественное закрепление пройденного материала.

Исходя из специфики изучаемого раздела и определенного умения, на формирование которого должны быть направлены задания можно сделать вывод о том, что в настоящем исследовании в дальнейшем будут рассматриваться аналитические, творческие и самообучающие задания.

В рамках специфики обучения программированию примерами аналитических задач являются: анализ созданного алгоритма с целью поиска ошибок, составление оптимального алгоритма из заранее готового избыточного набора программных блоков.

К творческим типам заданий относятся: переработка уже известного алгоритма с учетом предоставленного условия, на основе готового программного кода добиться аварийного завершения программы с определенным типом ошибки, разработка программы без определенного условия для ее работы лишь на основе набора входных и соответствующих выходных значений.

К самообучающим заданиям можно отнести: разработка программы, основанной на структурах программирования, еще не изученных на занятиях.

*Таким образом,* из представленных возможных типов заданий для формирования умения отладки можно использовать следующие типы заданий: анализ созданного алгоритма с целью поиска ошибок, на основе готового программного кода добиться аварийного завершения программы с определенным типом ошибки.

### **1.3. Проектирование методики формирования у обучающихся умения проведения отладки программного кода**

В настоящее время существует противоречие между необходимостью обучающимися умением проводить отладку программного кода для полного освоения раздела программирования и недостаточной разработанностью методики формирования данного умения. Поэтому возникает необходимость в проектировании методики формирования умения проводить отладку программы у обучающихся средней школы в процессе обучения программированию.

Цель проектирования – создание модели методики формирования умения отладки программы, реализация которой будет способствовать повышению результативности освоения раздела программирования среди учащихся средней школы.

На основе анализа содержания процесса поиска и устранения ошибок программного кода и дидактических возможностей выделим компоненты умения проводить отладку, которые будем развивать у обучающихся средней школы в процессе изучения раздела программирования:

- умение находить синтаксические ошибки и исправлять их;
- умение проводить тестирование программы и находить семантические ошибки, приводящие к аварийному завершению программы, и исправлять их;
- умение проводить локализацию логической ошибки на основании проведенного тестирования и исправлять ее.

Для установления уровня сформированности каждого из перечисленных действий нужно понимать, какие конкретные операции лежат в его основе. На основе анализа психолого-педагогической и методической литературы конкретизируем компоненты умений проведения отладки программного кода, выделив пооперационный состав каждого действия.

Таблица 1

## Характеристика умения проводить отладку программы

Компонент	Описание	Деятельностный состав
К1: умение находить синтаксические ошибки и исправлять их	аналитическая деятельность, направленная на объективную оценку собственного интеллектуального труда и проведения работы над ошибками	<ul style="list-style-type: none"> <li>- умение работать со средой разработки;</li> <li>- умение исправлять синтаксические ошибки;</li> <li>- умение выявлять наличие синтаксической ошибки до запуска программы</li> </ul>
К2: умение проводить тестирование программы и находить семантические ошибки, приводящие к аварийному завершению программы, и исправлять их	интеллектуальная деятельность, направленная на проработку всех возможных вариантов исхода алгоритма и анализа программы с целью поиска неверно использованных команд	<ul style="list-style-type: none"> <li>- умение определять область допустимых значений программы;</li> <li>- умение составлять наборы тестов для области допустимых значений, так и для случаев, выходящих за их пределы;</li> <li>- умение расшифровывать результаты тестирования в случаях аварийной остановки программы;</li> <li>- умение определять тип произошедшей ошибки на этапе запуска программы, исходя из выводимых программой данных;</li> <li>- умение анализировать алгоритм;</li> <li>- умение определять неверно использованные команды языка программирования и их исправление</li> </ul>
К3: умение проводить локализацию логической ошибки на основании проведенного тестирования и исправлять ее	интеллектуальная деятельность, направленная на поиск и исправление ошибок, допущенных на раннем этапе проработки алгоритма с последующим исправлением.	<ul style="list-style-type: none"> <li>- умение определять наличие ошибки логического типа на основе созданного набора тестов;</li> <li>- умение находить ошибки логического типа в коде;</li> <li>- умение видеть различные варианты решения задачи и переписывать неправильные блоки готового алгоритма</li> </ul>

На основании таблицы 1 сформулируем этапы формирования умения проводить отладку программы.

Первый этап: формирование умения находить синтаксические ошибки и исправлять их.

Второй этап: формирование умения проводить тестирование программы и находить семантические ошибки, приводящие к аварийному завершению программы, и исправлять их.

Третий этап: формирование умения проводить локализацию логической ошибки на основании проведенного тестирования и исправлять ее.

Для формирования умения проводить тестирование и исправление программного кода будем проводить контроль каждого определенного умения на различных вариантах заданий, что позволит наиболее точно определить сформированность всех умений.

*Низкий уровень:* у учащегося сформировано только умение находить синтаксические ошибки и исправлять их.

*Средний уровень:* у учащегося сформировано умение находить синтаксические ошибки и исправлять их и умение проводить тестирование программы и находить семантические ошибки, приводящие к аварийному завершению программы, и исправлять их.

*Высокий уровень:* у учащегося сформированы все умения осуществления отладки программы.



**Рис. 3** Уровни сформированности умения проводить отладку программы

Основываясь на итогах анализа компонентов методической системы обучения, предложенных в работах В.П. Монахова и А.М. Пышкало, под методикой формирования у учащихся умения проводить отладку программы в процессе изучения раздела программирования в процессе обучения информатике будем понимать совокупность взаимосвязанных и взаимообусловленных компонентов:

- целевого, включающего в себя описание деятельностного состава каждого компонента умения проводить отладку программы;
- содержательного, в котором приводятся темы раздела программирования, наиболее подходящие для формирования исследуемого умения и разнообразные типы заданий, позволяющие наиболее точно оценить сформированность умения проводить отладку программы.
- операционально-деятельностного, включающего в себя описание методов, средств и принципов обучения. Для формирования умения проводить отладку программы используются методы обучения:
  - групповая работа и индивидуальная работа;
  - методы рефлексии;
  - методы контроля и самоконтроля;

- практический метод.

Средствами обучения являются:

- среда разработки PyCharm;
- Online Test Pad;
- MS Excel.

Методика формирования умения проводить отладку программного кода основывается на принципах обучения: проблемное обучение, сознательность.

- контрольно-оценочного, выделяющего уровни сформированности умения с помощью диагностики, контроля (текущий, итоговый) и оценки результатов деятельности.

Так как формирование умения проводить отладку программного кода состоит из трех этапов, включающих в себя одинаковые компоненты, за исключением цели и результата, то модель процесса формирования умения осуществлять поиск и исправление ошибок в программе можно представить в виде схемы (Рис. 4).

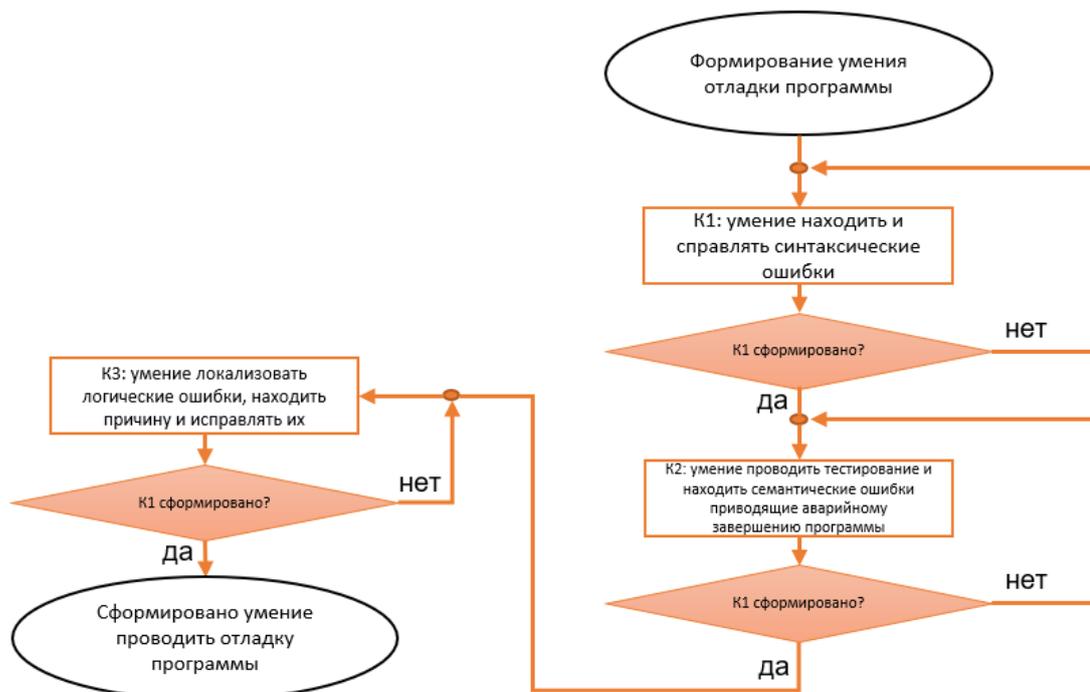


Рис. 4 Модель процесса формирования умения проводить отладку программы

Таким образом, разработанная методика формирования у учащихся умения проводить отладку программы состоит из трех взаимосвязанных этапов, на каждом выявленном этапе определены целевой, содержательный, операционально-деятельностный, контрольно-регулирующий и оценочно-результативный компоненты, определяющих возможность обучения учащихся на данном этапе.

## **Выводы по материалам главы 1**

1. Под отладкой программы будем понимать комплексный процесс, направленный на тестирование программы с целью обнаружения ошибок, их классифицирования на основе природы происхождения, локализации в программном коде с последующим устранением. Представляется целесообразным исследовать средства и способы формирования умения проводить тестирование и отладку программного кода на ранних стадиях обучения программированию с целью успешного освоения рассматриваемого раздела информатики

2. Проведенный анализ типов учебных заданий позволил выделить те задания, использование которых позволит достичь сформированности исследуемого умения проводить отладку программного кода: анализ созданного алгоритма с целью поиска ошибок, на основе готового программного кода добиться аварийного завершения программы с определенным типом ошибки.

3. Разработанная методика формирования у учащихся умения осуществлять отладку программы состоит из трех взаимосвязанных этапов, на каждом выделенном этапе выделены целевой, содержательный, операционально-деятельностный, контрольно-регулирующий и оценочно-результативные компоненты, определяющие возможность обучения учащихся на данном этапе.

## **Глава 2. Методика формирования у обучающихся умения выполнять отладку программы**

### **2.1. Компоненты методики формирования умения выполнять отладку программы**

В предыдущем параграфе были выделены три взаимосвязанных этапа процесса формирования у учащихся умения осуществлять учебное сотрудничество средствами облачных технологий. На первом этапе формируется компонент К1: умение находить синтаксические ошибки и исправлять их. Второй этап направлен на формирование компонента К2: умение проводить тестирование программы и находить семантические ошибки, приводящие к аварийному завершению программы, и исправлять их. Третий этап формирует компонент К3: умение проводить локализацию логической ошибки на основании проведенного тестирования и исправлять ее. Рассмотрим реализацию перечисленных этапов.

Реализация первого этапа начинается с освоения учащимися принципов работы в интерактивных средах разработки. Для этого на первых занятиях изучения раздела программирования учащиеся знакомятся с основными функциями определенного программного обеспечения и начинают систематически использовать его для выполнения учебных заданий. На начальных этапах изучения основной задачей для обучающихся является привыкнуть к использованию среды разработки и изучить основной минимум ее возможностей. Это достигается благодаря системе постепенно усложняющихся заданий, охватывающих различный функционал программы.

В качестве первых заданий целесообразно давать задания, основанные на основе изученных функций языка для отработки насмотренности синтаксиса и понимания принципов реагирования среды разработки на появление синтаксической ошибки.

Примером такого задания может служить практическая работа с мини-исследованием. Рассмотрим процесс выполнения практической работы на сложение двух переменных и последующим выводом результата.

*Методические рекомендации:*

1. Планирование такого задания осуществляется учителем, который ставит цели выполнения задания и предвидит результаты выполнения задания.

2. Учащимся предлагается написать алгоритм на сложение двух переменных на языке программирования Python в среде разработки PyCharm с сохранением всех различных результатов работы программы в виде скриншотов в текстовый редактор.

3. Учащиеся выполняют задание и делают скриншоты кода и консоли вывода после каждого запуска программы, вызванного изменениями программного кода.

4. После выполнения практической части задания учащимся под руководством учителя предлагается разобрать скриншоты и провести анализ причин, по которым некоторые алгоритмы привели к аварийному завершению программы, и разобрать подсказки среды разработки в таких случаях.

5. После выполнения всего учебного задания учитель подводит итоги, предлагает способы объективной самостоятельной оценки результатов учащихся.

На первом этапе учитель постепенно обучает учащихся объективно оценивать свою работу и не бояться делать ошибки, анализировать написанный алгоритм на предмет возможных синтаксических ошибок и способами их устранения.

Главной целью на первом этапе является научить учащихся работать со средой разработки, обращать внимание на подсказки среды разработки и

правильно их интерпретировать при программировании, объективно оценивать и делать выводы результатов своей работы.

В конце первого этапа необходимо провести диагностику сформированности умения находить синтаксические ошибки и исправлять их. По завершению первого этапа учащиеся должны:

- понимать цели учебного задания;
- уметь работать со средой разработки;
- анализировать программный код и видеть большинство синтаксических ошибок до запуска программы;
- уметь исправлять синтаксические ошибки;
- уметь отказаться от собственного решения в пользу оптимального.

Если учащийся может выполнить больше половины действий, то умение К1 сформировано, если меньше половины – умение К1 не сформировано. Учащиеся, у которых умение К1 не сформировалось повторно проходят первый этап на других заданиях.

Действия лежащие в основе формирования умения К1 являются основными для формирования умения осуществлять отладку программного кода и базовыми для освоения раздела программирования. Следовательно, первый этап является самым главным, так как на нем закладываются основные умения работы с программой.

Реализация второго этапа начинается с актуализации первого, но теперь поиск и исправление синтаксических ошибок учащиеся осуществляют самостоятельно при выполнении любого задания на программирование. На этом этапе формируется компонент К2: умение проводить тестирование программы и находить семантические ошибки, приводящие к аварийному завершению программы, и исправлять их.

Задача учителя научить проводить тестирование программы, правильно интерпретировать результаты и определять тип ошибки.

На данном этапе ученик должен хорошо определять данные, которые программа должна обрабатывать и вручную рассчитывать результат, который должна вывести программа. По сравнению с этим определение значений, которые выходят за область допустимого, является более сложной задачей для учащегося и на это следует обращать больше внимания. Так как случаи появления семантических ошибок зачастую связаны именно с отсутствием или неправильной обработкой результатов, которые не предусмотрены условием задачи. Локализация семантических ошибок осуществляется на базе умения работать с консолью программы в среде разработки.

Рассмотрим процесс выполнения практического задания, направленный на формирование умения определять область допустимых значений программы вычисления дискриминанта квадратного уравнения по его коэффициентам и составлять набор данных для ее тестирования.

*Методические рекомендации:*

1. Задания, направленные на отработку умения проводить тестирование могут применяться в двух форматах: самостоятельно и как составная часть задания на разработку программы по условию. В любом из этих случаев учащиеся должны иметь уже написанную в каком-то виде программу, корректность кода которой необходимо проверить.

2. Учащимся предлагается проанализировать начальное условие задачи программирования на определение типов данных, на основе которых программа должна осуществлять свою работу: целые числа, вещественные числа или строки. На основе проведенного анализа учащимся нужно определить какого формата входные данные программа должна обрабатывать (область допустимых значений), а какие должны приводить к завершению ее работы или игнорироваться (область недопустимых значений).

3. Учащиеся случайным образом составляют массив входных данных выделенного типа и записывают их в текстовый файл в удобном для них формате. После чего необходимо без участия проверяемой программы рассчитать эталонный результат выполнения программы и сопоставить их со сгенерированными входными данными.

4. Аналогичным образом учащиеся составляют набор тестовых данных, которые потенциально недопустимы при выполнении программы, и сформулировать выражения, которые должна выводить программа в случаях их ввода.

5. После чего учащиеся проводят тестирование программы путем ее запуска, ввода значений из составленного набора и сравнения полученного результата с эталонным. В случае аварийного завершения работы программы, делают себе пометки для дальнейшего поиска ошибки.

6. По результатам тестирования исходя из пометок учащимся предлагается локализовать полученные ошибки и исправить их.

7. После выполнения всего учебного задания учитель подводит итоги, предлагает способы объективного оценивания результатов учащихся.

К концу второго этапа учащиеся должны:

- уметь определять области допустимых и недопустимых значений программы;
- составлять наборы данных для проведения тестирования программы;
- определять местоположение и причины возникновения семантических ошибок и исправлять их.

По завершению второго этапа проводится диагностика сформированности умения К2. Если учащийся может выполнить больше половины действий, то умение К2 сформировано, если меньше – умение К2 не сформировано. Учащиеся, у которых не сформировалось повторно проходят первый этап на других заданиях.

Реализация третьего этапа начинается с актуализации первого и второго этапов. Учащиеся самостоятельно выполняют все действия, относящиеся к компонентам К1 и К2.

Особое внимание на данном этапе отводится интерпретации результатов тестирования отличных от эталонных, но не приводящих к аварийному завершению работы программы и анализу программы для определения точного местоположения логической ошибки.

Перечень причин возникновения логической ошибки в программе очень разнообразен. Именно из-за этого локализация и исправление ошибок этого типа является самой сложной задачей при отладке программы.

Существует много принципов локализации логических ошибок, основанных на принципах анализа программного кода. Для нахождения ошибок при изучении программирования на уровне среднего общего образования нет необходимости в их погружение из-за их сильной специфики, поэтому предлагаем для изучения два выделенных метода: метод аварийной печати и метод пошагового выполнения программы.

Для использования метода аварийной печати обучающиеся должны хорошо понимать структуру работы своей программы. Он заключается в добавлении команды печати в консоль (в рамках исследования рассматривается язык программирования python с командой print) в ключевые моменты алгоритма: вход в структуру цикла или проверки условия, изменение значения переменной, вызов функции, в структуру обработки ошибок. После чего снова проводится тестирование программы с анализом получаемых значений и отслеживанием изменений выводимых данных. При использовании данного метода нужно грамотно выбирать печатаемые данные, чтобы не запутаться в собственном способе поиска ошибок. Следует использовать минимальное возможное количество добавлений команды печати, которые смогут дать учащимся объективную картину работу программы. После нахождения момента неправильной

работы программы нужно просто найти строку программного кода с некорректным выводом и понять причину неверного вычисления или неправильно написанного условия.

Метод пошагового выполнения программы не требует добавления новых команд в программу, но основан на хорошем уровне умения работать со средой разработки. При его применении ставится точка остановки программы в строке, недалеко от предполагаемого места ошибки и запускается режим пошагового выполнения работы программы (режим debugging). После запуска программы в данном режиме программа до точки остановки выполняется в стандартном режиме с последующим переключением на ручное управление шагами. Среда разработки показывает изменение всех переменных присутствующих в программе и выделением выполняемой строки – необходимо просто грамотно отследить изменение интересующей переменной или корректность проверяемого условия, присутствующего в программе.

После успешной локализации ошибки любым из представленных методов учащемуся необходимо переработать структуру алгоритма для исправления ошибки исходя из причины возникновения ошибки и переписать некорректный блок кода.

Важно после исправления ошибки повторно провести тестирование программы для того, чтобы убедиться в том, что ошибка была исправлена. В случае, когда ошибка не была исправлена снова возвращаемся к процессу локализации ошибки и подробному анализу разработанного алгоритма.

В процессе отладки самым длительным является процесс исправления именно логических ошибок, так как они зачастую неочевидны и процесс локализации, и исправления может повторяться множество раз.

*Таким образом,* соблюдение предложенной последовательности этапов является необходимым условием для успешного формирования у учащихся

умения проводить отладку программы при изучении раздела программирования в процессе изучения информатики.

## 2.2. Учебные задания для формирования умения выполнять отладку программы

В основе формирования умения осуществлять отладку программного кода лежит применение совокупности специальных учебных заданий. В ходе исследования выявлено, что навык отладки формируется путем решения различных типов заданий, покрывающих различные аспекты исследуемого умения. Рассмотрим различные типы учебных заданий, применяемых на каждом этапе формирования умения проводить поиск и исправления ошибок программы и сформулируем методические рекомендации по их использованию.

**Этап 1** (формирование К1: умение находить синтаксические ошибки и исправлять их)

Учебное задание «Определение синтаксических ошибок».

Тип задания: практическая работа.

Проанализируйте представленную программу на наличие синтаксических ошибок. Для каждой найденной ошибки заполните строку таблицы.

```

1 a = 7
2 b = int(input())
3 b = a - v
4 prit(b)
5

```

Номер строки с ошибкой	Причина возникновения ошибки	Корректный вариант строки

*Методические рекомендации:*

Задания данного формата могут применяться на основе любой темы раздела программирования с изучением функций или управляющих конструкций.

Данный тип заданий направлен на визуальный поиск синтаксических ошибок по подсказкам среды разработки. После нахождения ошибки ученик заполняет номер строки с ошибкой, предполагаемую им причину ошибки и как правильно должна выглядеть эта строка после ее исправления.

Учебное задание «Локализация синтаксической ошибки по выходным данным консоли».

Тип задания: практическая работа.

Некая программа после запуска была аварийно завершена и в консоли вывода появились следующие сведения о случившейся ошибке.

```
File "C:\DATA\Python_project\fun\cyber\main.py", line 2
  b = int(input()
      ^
SyntaxError: '(' was never closed

Process finished with exit code 1
```

Исходя из полученного вывода, определите номер строки, в которой произошла ошибка, причину возникновения ошибки, и корректный вариант исправления ошибки.

*Методические рекомендации:*

Данный тип заданий направлен на обучение учащихся работать с консолью программы и исходя из ее вывода проводить локализацию и исправление синтаксических ошибок.

Учебное задание «Преднамеренное создание синтаксических ошибок».

Тип задания: практическая работа.

```

1 a = int(input())
2 b = int(input())
3 print('Площадь прямоугольника равна', a*b)

```

Напишите примеры синтаксических ошибок, которые могут появиться в третьей строке при программировании данной задачи

*Методические рекомендации:*

Задания данного формата могут применяться на основе любой темы раздела программирования. При проектировании подобных задач важно точно определить тип и местонахождение ошибки, которую учащиеся должны создать.

Данный тип заданий предназначен для переключения учащихся на деструктивный тип мышления, который необходим для повышения результативности поиска ошибок в программном коде, и для расширения кругозора возможных синтаксических ошибок. Чем больше примеров возможных ошибок учащиеся встречают на ранних этапах изучения программирования, тем проще в будущем проводить отладку ошибок подобного типа.

**Этап 2** (умение проводить локализацию логической ошибки на основании проведенного тестирования и исправлять ее)

Учебное задание «Разработка набора тестовых данных».

Тип задания: практическая работа.

Проведите тестирование программы для вычисления суммы двух чисел на основе приведенного набора и по его результатам заполните таблицу

№	Входные данные		Эталонный результат	Результат работы программы	Комментарии
1	-3	6	3		
2	-5	-8	-12		
3	0	14	14		
4	3	42	45		

5	-36	0	-36		
---	-----	---	-----	--	--

*Методические рекомендации:*

Данная задача выполняется сразу после решения задачи на программирование программы с указанным условием, в данном случае сумма двух чисел.

Задания такого формата направлены на отработку умения составлять и работать с наборами тестовых данных. В зависимости от уровня учащихся задания могут сильно варьироваться: набор может быть задан изначально или его необходимо продумать, можно вносить данные только из области допустимых значений программы в самом начале изучения программирования или предусматривать и заведомо ложные тесты. Столбец «Результат работы программы» ученики заполняют согласно получившемуся выводу программы при вводе указанных значений. После чего значения данного столбца сравниваются со столбцом «Эталонный результат» и при несовпадении заполняется столбец «Комментарий» возможной причиной ошибки.

**Этап 3** (умение проводить локализацию логической ошибки на основании проведенного тестирования и исправлять ее)

Учебное задание «Проведение отладки готовой программы».

Тип задания: практическая и исследовательская работа.

**Разработайте программу, согласно условию и проведите отладку программы**

Даны два натуральных числа  $m$  и  $n$  ( $m \leq n$ ). Напишите программу, которая выводит все числа от  $m$  до  $n$  включительно удовлетворяющие **хотя бы одному** из условий:

- число кратно 17;
- число оканчивается на 9;
- число кратно 3 и 5 одновременно.

**Формат входных данных**

На вход программе подаются два натуральных числа  $m$  и  $n$  ( $m \leq n$ ), каждое на отдельной строке.

**Формат выходных данных**

Программа должна вывести числа в соответствии с условием задачи.

*Методические рекомендации:*

Данный тип заданий являются завершающими в формировании умения проводить отладку программного кода. Они представляют собой совмещение всех этапов программирования: как создание алгоритма и его перевода на язык программирования высокого уровня, так и полный цикл отладки готовой программы. Учащимся необходимо переключаться с конструктивного на деструктивный тип мышления, что является одной из самых сложных задач даже для практикующих программистов. В данном формате заданий сложность задачи напрямую зависит от условия для разработки программы.

Таким образом, приведенные специальные учебные задания для каждого этапа методики позволят сформировать у учащихся умение осуществлять отладку программы при изучении раздела программирования на уроках информатики.

### **2.3. Организация опытно-поисковой работы и ее результаты**

Целью опытно-поисковой работы являлась проверка научной гипотезы исследования о результативности предлагаемой методики формирования умения осуществлять отладку программного кода в процессе изучения раздела программирования на уроках информатики.

Задачи опытно-поисковой работы заключались в следующем:

1. Изучение состояния проблемы организации учебного сотрудничества при обучении школьников информатике с использованием облачных технологий.

2. Выявление положительных тенденций применения разработанной методики при ее апробации.

3. Экспертная оценка разработанной методики на предмет целесообразности и результативности при формировании умения проводить отладку программного кода.

В качестве экспертов выступили: учитель информатики лицея №110, преподаватель информационных технологий детского технопарка «Кванториум» СвДЖД, преподаватель гимназии №8. Экспертам была предложена анкета, состоящая из следующих вопросов:

1. Соответствие уровня заданий для начального уровня освоения программирования для учащихся средней школы.
2. Степень влияния разработанных заданий для формирования компонентов умения отладки программного кода.
3. Обеспечения формирования умения проводить отладку программы посредством разработанного комплекса учителем.
4. Уровень проработки методических рекомендаций по использованию разработанного комплекса учителем.
5. Рекомендации экспертов.

Анализ результатов показал, что разработанная методика полностью соответствует содержанию раздела программирования при изучении информатики в средней школе и в полном объеме способствует формированию умения проводить отладку программы. Задания, разработанные для методики, можно использовать как для непосредственно формирования указанного умения, так и для тренажа на этапе повторения основных действий при проведении тестирования программы и поиска ошибок с их последующим устранением.

*Таким образом,* проведенная экспертная оценка свидетельствует о результативности разработанной методики для формирования умения проводить отладку программного кода и может применяться при изучении программирования на уроках информатики в средней школе.

## Выводы

Сопоставление результатов работы с поставленными задачами позволяет заключить следующее:

1) На основании анализа научно-педагогической и методической литературы уточнено содержание понятия «отладки программного кода», выделены элементы умения учащегося проводить отладку программы.

2) На основании анализа дидактических возможностей формирования умения проводить отладку программы у учащихся средней школы были выделены типы заданий, которые гарантируют сформированность элементов исследуемого умения.

3) На основе выделенных этапов формирования умения осуществлять отладку программы (первый этап: формирование умения находить синтаксические ошибки и исправлять их, второй этап: формирование умения проводить тестирование программы и находить семантические ошибки, приводящие к аварийному завершению программы, и исправлять их, третий этап: формирование умения проводить локализацию логической ошибки на основании проведенного тестирования и исправлять ее) построена структурно-функциональная модель методики, в которой на каждом выявленном этапе определены целевой, содержательный, операционно-деятельностный, контрольно-регулирующий и оценочно-результативный компоненты, определяющих возможность обучения учащихся на каждом этапе.

4) Проведенная опытно-поисковая работа по проверке результативности применения методики формирования у учащихся умения осуществлять отладку программного кода в процессе изучения раздела программирования на уроках информатики позволяет сделать однозначный вывод о том, что применение предложенной в работе методики обеспечивает формирование у учащихся умения проводить отладку программного кода. Тем самым полностью подтверждается гипотеза исследования.

Апробация результатов работы путем экспертной оценки показала высокую результативность формирования у учащихся умения поиска и исправления ошибок в программе.

Таким образом, следует считать, что задачи исследования полностью выполнены, цель достигнута.

## Источники информации

### Примеры правильного оформления ссылок по ГОСТ- Р 7.0.100-2018

1. Лапчик М П., Семакин И. Г. , Хеннер Е. К., Методика преподавания информатики [лежит в папке]
2. <https://monographies.ru/en/book/section?id=4632>]
3. [https://bigenc.ru/technology\\_and\\_technique/text/2697971](https://bigenc.ru/technology_and_technique/text/2697971)
4. [https://studopedia.ru/3\\_175688\\_otladka-programmnogo-obespecheniya.html](https://studopedia.ru/3_175688_otladka-programmnogo-obespecheniya.html)
5. <https://testirovanieprob.blogspot.com/p/33.html>
6. <https://habr.com/ru/post/549054/>]
7. <https://studfile.net/preview/7010475/page:4/>]
8. ФГОС ОСО
9. <https://monographies.ru/en/book/section?id=4632>
10. Основы тестирования программного обеспечения. К. А. Кулкаков
11. <https://studfile.net/preview/775279/page:8/>
12. УМК Семакин
13. УМК Босова

## **Приложения**

### ***Приложение 1***

Приложения содержат только информационные материалы. Все пояснения к ним должны содержаться в основном тексте.

**Приложение 2**

Каждое Приложение располагается с новой страницы.