

Городское управление образованием администрации города Черногорска  
«Основная общеобразовательная школа № 6» - филиал Муниципального  
бюджетного общеобразовательного учреждения «Средняя общеобразовательная  
школа № 7 имени Героя Советского Союза Петра Акимовича Рубанова»

**Байкалова Мария Сергеевна,  
учитель математики и информатики**

**АВТОМАТИЗАЦИЯ ПЕДАГОГИЧЕСКОЙ ДИАГНОСТИКИ:  
ИСПОЛЬЗОВАНИЕ PYTHON, EXCEL И ЯНДЕКС ФОРМ ДЛЯ АНАЛИЗА  
ПРОБЕЛОВ В ЗНАНИЯХ**

Черногорск 2025

## Оглавление

Введение .....	3
1. Теоретические основы автоматизации диагностики .....	4
2. Практическая реализация системы автоматизации .....	5
2.1. Использование Яндекс Форм для сбора данных .....	5
2.2. Обработка данных в Excel .....	5
2.3. Автоматизация анализа с помощью Python .....	5
3. Реализация системы в практической деятельности .....	6
Заключение .....	7
Список использованных источников .....	7
Приложения .....	8

## Введение

**Актуальность темы:** в условиях современной школы с высокой учебной нагрузкой учитель сталкивается с необходимостью оперативного анализа образовательных результатов большого количества учащихся. Традиционные методы проверки и анализа работ отнимают значительное время, что снижает эффективность педагогического вмешательства. Автоматизация процессов диагностики позволяет не только экономить временные ресурсы педагога, но и обеспечивать более глубокий и объективный анализ пробелов в знаниях учащихся.

**Цель:** разработать и внедрить систему автоматизированную педагогическую диагностику для оперативного выявления пробелов в знаниях учащихся и построения индивидуальных образовательных траекторий.

**Задачи:**

1. Проанализировать возможности современных цифровых инструментов для автоматизации диагностики.
2. Разработать алгоритм взаимодействия Яндекс Форм, Excel и Python для анализа образовательных результатов.
3. Апробировать систему автоматизации в практической деятельности учителя.
4. Оценить эффективность предложенной системы для повышения качества образования.

## 1. Теоретические основы автоматизации диагностики

Автоматизация педагогической диагностики представляет собой процесс использования цифровых инструментов для сбора, обработки и анализа данных об образовательных результатах учащихся. В основе системы лежит принцип оперативной обратной связи, когда учитель получает структурированную информацию о знаниях учащихся в кратчайшие сроки после проведения диагностической работы.

Ключевые преимущества автоматизированной системы:

- **Оперативность** — анализ результатов проводится в течение минут после завершения работы учащимися.
- **Объективность** — исключается субъективный фактор при проверке и анализе.
- **Детализация** — возможность анализировать результаты по отдельным темам и заданиям.
- **Визуализация** — автоматическое создание графиков и диаграмм для наглядного представления данных.

Современные образовательные стандарты требуют от учителя реализации индивидуального подхода, что невозможно без эффективной системы диагностики, позволяющей своевременно выявлять образовательные дефициты.

## 2. Практическая реализация системы автоматизации

### 2.1. Использование Яндекс-Форм для сбора данных.

Яндекс-Формы применяются для создания цифровых версий диагностических работ. Преимущества данного инструмента:

- простота создания и настройки
- автоматический сбор ответов в таблицу
- возможность ограничения времени выполнения работы
- интеграция с другими сервисами.

**Практический пример:** При изучении темы «Алгоритмизация и программирование» создается форма с заданиями различного уровня сложности — от тестовых вопросов до задач с открытым ответом (Приложение 1).

### 2.2. Обработка данных в Excel.

Полученные из Яндекс Форм данные импортируются в Excel для первичной обработки:

- сортировка результатов по классам и учащимся
- расчет процентного выполнения по каждому заданию
- выявление заданий с наименьшим процентом выполнения
- формирование сводных таблиц для анализа по темам.

**Автоматизация в Excel** осуществляется с помощью формул и условного форматирования, что позволяет визуально выделять проблемные зоны (Приложение 2).

### 2.3. Автоматизация анализа с помощью Python.

Для углубленного анализа используется Python с библиотеками pandas, matplotlib и seaborn:

```
python
import pandas as pd
import matplotlib.pyplot as plt
```

```
# Загрузка данных из Excel
df = pd.read_excel('diagnostics_results.xlsx')

# Анализ результатов по темам
topic_analysis = df.groupby('topic')['score'].mean()

# Визуализация результатов
plt.figure(figsize=(10, 6))
topic_analysis.plot(kind='bar')
plt.title('Результаты диагностики по темам')
plt.ylabel('Средний балл')
plt.savefig('topic_analysis.png')
```

Данный скрипт позволяет автоматически:

- группировать результаты по темам
- выявлять наиболее проблемные темы
- создавать наглядные визуализации для работы с учащимися и родителями.

### **3. Реализация системы в практической деятельности**

Внедрение системы автоматизированной диагностики было апробировано в 8-9 классах (общее количество учащихся — 67 человек) в течение 2024-2025 учебного года.

#### **Этапы реализации:**

1. Создание банка диагностических работ по основным темам курса информатики.
2. Проведение входной, промежуточной и итоговой диагностики.
3. Автоматизированная обработка результатов.
4. Формирование индивидуальных рекомендаций для учащихся.
5. Корректировка учебного процесса на основе полученных данных.

#### **Достигнутые результаты:**

- сокращение времени на проверку и анализ работ на 70%
- повышение средней успеваемости по предмету на 15%
- повышение удовлетворенности учащихся и родителей объективностью оценивания.

#### **Рекомендации по использованию:**

1. Начните с автоматизации одной-двух диагностических работ.
2. Постепенно расширяйте банк заданий.
3. Используйте визуализации для обсуждения результатов с учащимися.
4. Интегрируйте систему с электронным журналом.
5. Обучите коллег использованию инструментов автоматизации.

## **Заключение**

Разработанная система автоматизированной педагогической диагностики доказала свою эффективность в практической деятельности. Она позволяет не только экономить время учителя, но и обеспечивает качественно новый уровень анализа образовательных результатов.

### **Перспективы работы:**

1. Разработка мобильного приложения для учащихся с автоматическим формированием индивидуальных заданий
2. Создание системы прогнозирования образовательных результатов на основе анализа больших данных
3. Разработка межпредметной системы диагностики для использования учителями разных предметных областей
4. Внедрение автоматизированных систем диагностики является необходимым условием для повышения качества образования в современной школе и реализации требований ФГОС.

## **Список использованных источников**

1. Федеральный государственный образовательный стандарт основного общего образования.
2. Федеральная основная образовательная программа основного общего образования.
3. Асмолов А.Г. «Формирование универсальных учебных действий в основной школе».
4. Официальная документация по Python библиотекам pandas, matplotlib.
5. Методические материалы по работе с Яндекс Формами и Excel.

## ПРИЛОЖЕНИЕ 1

**Пример диагностической работы в Яндекс – Формах.**

**Тема:** «Алгоритмизация и программирование» (8 класс)

**Ссылка на форму:** [https://forms.yandex.ru/cloud/пример\\_диагностики\\_алгоритмы](https://forms.yandex.ru/cloud/пример_диагностики_алгоритмы)

**Структура диагностической работы:**

1. **Блок А. Базовые понятия** (тестовые задания с выбором одного ответа)
  - о Вопрос: Что такое алгоритм?
    - Варианты:
      - а) Программа на языке Python
      - б) Последовательность действий для решения задачи ✓
      - в) Математическая формула
      - г) Устройство компьютера
    - о Вопрос: Какое свойство алгоритма означает, что он состоит из конкретных шагов?
      - Варианты:
        - а) Дискретность ✓
        - б) Массовость
        - в) Детерминированность
        - г) Понятность
  - 2. **Блок В. Анализ алгоритмов** (задания на установление соответствия)
    - о Установите соответствие между блок-схемами и их описаниями
  - 3. **Блок С. Практическое задание** (задание с открытым ответом)
    - о "Составьте алгоритм для нахождения суммы всех четных чисел от 1 до 100. Запишите его в виде блок-схемы или на языке Python"

**Настройки формы:**

- Ограничение времени: 30 минут
- Обязательность ответов на все вопросы
- Автоматический сбор email учащихся для идентификации
- Настройка показа результатов после завершения

## ПРИЛОЖЕНИЕ 2

**Шаблон таблицы Excel для анализа результатов.**

**Структура таблицы «Результаты\_диагностики.xlsx»:**

A	B	C	D	E	F	G	H	I
ФИО	Класс	Вопрос 1	Вопрос 2	Вопрос 3	Блок A_балл	Блок B_балл	Блок C_балл	Итогов ый балл
Иванов А.	8А	1	1	0	8	5	3	16
Петрова Б.	8А	1	0	1	6	3	5	14

**Формулы для автоматического расчета:**

```
excel
=СУММ(С2:Е2) // Сумма баллов за блок А
=СРЗНАЧ(F2:F35) // Средний балл по классу за блок А
=СЧЁТЕСЛИ(Е2:Е35,<5") // Количество учащихся с низким результатом
```

#### **Условное форматирование:**

- Красный цвет: балл < 50% от максимального
- Желтый цвет: балл 50-80%
- Зеленый цвет: балл > 80%

#### **Сводная таблица для анализа по темам:**

- Группировка по классам
- Средний балл по каждому заданию
- Процент выполнения по темам

## **ПРИЛОЖЕНИЕ 3**

### **Примеры кода на Python для автоматизации анализа**

#### **1. Основной скрипт анализа результатов**

```
python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Загрузка данных из Excel
df = pd.read_excel('Результаты_диагностики.xlsx')

# Анализ результатов по классам
class_analysis = df.groupby('Класс')['Итоговый_балл'].agg(['mean', 'std', 'count'])
print("Результаты по классам:")
print(class_analysis)

# Визуализация распределения баллов
plt.figure(figsize=(12, 6))

plt.subplot(1, 2, 1)
df.boxplot(column='Итоговый_балл', by='Класс')
plt.title('Распределение баллов по классам')
plt.suptitle("")

plt.subplot(1, 2, 2)
topic_means = df[['Блок_А_балл', 'Блок_В_балл', 'Блок_С_балл']].mean()
topic_means.plot(kind='bar', color=['lightcoral', 'lightblue', 'lightgreen'])
plt.title('Средние баллы по тематическим блокам')
plt.ylabel('Средний балл')

plt.tight_layout()
plt.savefig('анализ_результатов.png', dpi=300)
plt.show()
```

#### **2. Скрипт для выявления пробелов в знаниях**

```
python
```

```
# Анализ проблемных заданий
```

```
def analyze_problem_areas(df, threshold=0.6):
```

```
    """
```

```
    Выявляет задания с низким процентом выполнения
```

```
    threshold - порог успешности (0.6 = 60%)
```

```
    """
```

```
    problem_areas = {}
```

```
# Анализ по отдельным вопросам
```

```
for question in ['Вопрос1', 'Вопрос2', 'Вопрос3']:
```

```
    success_rate = df[question].mean()
```

```
    if success_rate < threshold:
```

```
        problem_areas[question] = success_rate
```

```
# Анализ по тематическим блокам
```

```
for block in ['Блок_A_балл', 'Блок_B_балл', 'Блок_C_балл']:
```

```
    max_score = df[block].max()
```

```
    success_rate = df[block].mean() / max_score
```

```
    if success_rate < threshold:
```

```
        problem_areas[block] = success_rate
```

```
return problem_areas
```

```
# Запуск анализа
```

```
problems = analyze_problem_areas(df)
```

```
print("Проблемные зоны:")
```

```
for area, rate in problems.items():
```

```
    print(f'{area}: {rate:.1%}')
```

### 3. Генератор индивидуальных отчетов

```
python
```

```
# Создание индивидуальных рекомендаций
```

```
def generate_student_report(row):
```

```
    report = f'Отчет по диагностике для {row["ФИО"]}\n'
```

```
    report += f'Итоговый балл: {row["Итоговый_балл"]}\n\n'
```

```
# Рекомендации по блокам
```

```
if row['Блок_A_балл'] < 5:
```

```
    report += "• Рекомендуется повторить базовые понятия алгоритмизации\n"
```

```
if row['Блок_B_балл'] < 4:
```

```
    report += "• Необходима практика анализа готовых алгоритмов\n"
```

```
if row['Блок_C_балл'] < 3:
```

```
    report += "• Требуется дополнительная работа по составлению алгоритмов\n"
```

```
return report
```

```
# Создание отчетов для всех учащихся
```

```
for index, row in df.iterrows():
```

```
report = generate_student_report(row)
print(report)
print("-" * 50)
```

## ПРИЛОЖЕНИЕ 4

## Визуализации результатов диагностики

## 1. Диаграмма распределения результатов по классам

\*Рисунок 1. Сравнительный анализ результатов диагностики в параллели 8-х классов\*

## 2. Heatmap успеваемости по темам

python

## # Создание тепловой карты успеваемости

```
plt.figure(figsize=(10, 8))
```

```
topic_data = df[['Блок_A_балл', 'Блок_B_балл', 'Блок_C_балл']]  
topic_data_normalized = topic_data / topic_data.max()
```

```
sns.heatmap(topic_data_normalized,  
            cmap='RdYlGn',  
            annot=True,  
            fmt='.1%',  
            cbar_kws={'label': 'Процент выполнения'})
```

```
plt.title('Тепловая карта успеваемости по тематическим блокам')
plt.savefig('heatmap_успеваемость.png', dpi=300)
```

### 3. График динамики улучшения результатов

Пример данных для демонстрации эффективности системы:

Период	Средний балл	% успевающих
Сентябрь 2024	12.3	45%
Декабрь 2024	15.7	68%
Март 2025	17.2	82%

#### 4. Пример индивидуальной карты учащегося

text

## ИДИВИДУАЛЬНАЯ КАРТА УЧАЩЕГОСЯ

ФИО: Иванов Алексей

Класс: 8А

## РЕЗУЛЬТАТЫ ДИАГНОСТИКИ:

- Блок А: 7/10 баллов (70%) -  норма
  - Блок В: 4/8 баллов (50%) -  требуется внимание
  - Блок С: 2/6 баллов (33%) -  необходима коррекция

**РЕКОМЕНДАЦИИ:**

1. Повторить тему "Анализ блок-схем"
2. Выполнить дополнительные задания на составление алгоритмов
3. Обратиться к материалам урока от 15.03.2025

**Рекомендации по использованию приложений:**

1. **Для начала работы:** Используйте Приложение 1 как шаблон для создания своих диагностических работ.
  2. **Для быстрого анализа:** Загрузите данные в шаблон Excel из Приложения 2
  3. **Для углубленного анализа:** Запустите скрипты Python из Приложения 3
  4. **Для наглядности:** Используйте визуализации из Приложения 4 для педсоветов и родительских собраний
- Все материалы готовы к использованию и могут быть адаптированы под конкретные учебные задачи и предметные области.