

Министерство образования Республики Беларусь
Учреждение образования «Белорусский государственный университет
информатики и радиоэлектроники»

Институт информационных технологий

Кафедра информационных систем и технологий

**А. И. Парамонов, К. А. Павлюченко, А. Г. Савенко,
И. И. Фролов, И. А. Середа**

**МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ
ПО ВЫПОЛНЕНИЮ КУРСОВОЙ РАБОТЫ
ПО УЧЕБНОЙ ДИСЦИПЛИНЕ
«ПРИКЛАДНЫЕ ИНТЕЛЛЕКТУАЛЬНЫЕ
СИСТЕМЫ»**

*Для студентов 4 курса специальности
6-05-0612-01 «Программная инженерия»*

Минск БГУИР 2026

СОДЕРЖАНИЕ

Введение	3
1 Порядок выполнения курсовой работы	4
2 Требования к учебному элементу курсовой работы	6
2.1 Технические требования	6
2.2 Обязательные для реализации этапы курсовой работы	6
2.3 Дополнительные этапы выполнения курсовой работы	7
3 Структура пояснительной записки курсовой работы	10
3.1 Общая структура пояснительной записки курсовой работы	10
3.2 Требования к отдельным разделам пояснительной записки	10
4 Требования к оформлению пояснительной записки курсовой работы	13
Список использованных источников и рекомендованной литературы	14

ВВЕДЕНИЕ

*«Если ИИ и уничтожит человечество,
то только потому, что мы построили
в него наши собственные ошибки»*

Джордж Мартин, писатель

Учебным планом специальности 6-05-0612-01 «Программная инженерия» по учебной дисциплине «Прикладные интеллектуальные системы» (ПИС) предусмотрено выполнение курсовой работы.

Курсовая работа – это учебная работа, содержащая результаты теоретических и (или) экспериментальных исследований по отдельной учебной дисциплине и включающая совокупность аналитических, расчётных, исследовательских, оценочных заданий, объединённых общностью рассматриваемого объекта и предполагающая выполнение отдельных элементов конструкторских, технологических, программных, организационно-управленческих и других работ [1].

Цель курсовой работы по дисциплине – разработка прототипа прикладной интеллектуальной системы для решения реальной отраслевой задачи с применением современных технологических стеков (генеративные LLM, компьютерное зрение, рекомендательные системы, обнаружение аномалий) и закрепление навыков проектирования, разработки и промышленной эксплуатации ИИ-систем.

Курсовая работа является логическим завершением сквозной проектной деятельности, охватывающей тематические разделы 2 и 3 содержания учебного материала дисциплины, и расширяет результаты лабораторных работ за счёт дополнительного анализа, документирования и обоснования принятых инженерных решений, а также интеграции всех компонентов в единое работоспособное решение.

1 ПОРЯДОК ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ

1.1 Студенты выбирают тему курсовой работы из предложенного перечня тем (см. Приложение А) либо согласует с руководителем собственный вариант темы.

Тема формируется как сочетание трёх элементов:

- *предметная область* (например, техподдержка, e-commerce, медицина);
- *прикладная задача* (например, категоризация обращений, извлечение данных, рекомендации);
- *архитектурный подход к решению* (например, классический ML, fine-tuning encoder-модели, RAG, zero/few-shot LLM).

1.2 Курсовая работа может выполняться *командами по 2–3 студента* с распределением ролей в соответствии с выбранной тематикой. Ролевая модель в команде согласовывается с руководителем на этапе утверждения темы и до выдачи задания.

Пример возможных ролей:

- *инженер данных* – отвечает за сбор, подготовку и валидацию данных, построение пайплайнов обработки;
- *инженер машинного обучения* – отвечает за выбор, адаптацию и оценку качества модели;
- *инженер по интеграции и развёртыванию (при соответствующей реализации)* – отвечает за разработку REST API, контейнеризацию, создание веб-интерфейса и организацию мониторинга.

Работа может быть выполнена *индивидуально одним студентом*.

1.3 Студенты получают у руководителя курсовой работы лист задания с краткой постановкой задачи, описанием ожидаемых результатов и календарным планом работы (см. Приложение Б).

1.4 Студенты выполняют курсовую работу самостоятельно (не привлекая сторонней помощи, кроме своей команды), посещая плановые консультации (в том числе с применением ИКТ) согласно календарному плану.

1.5 Работу необходимо выполнить к установленной дате и в срок сдать руководителю указанный комплект:

- пояснительная записка (рекомендуемый формат «.docx» MS Word);
- программный проект (включая Git-репозиторий);
- графические материалы;
- презентация (рекомендуемый формат «.pptx» MS PowerPoint).

1.6 Студенты проходят защиту перед специально назначенной комиссией из числа преподавателей кафедры, во время которой демонстрируется работа прототипа. Каждый участник команды (при работе в команде) представляет свою часть работы и отвечает на вопросы комиссии индивидуально.

1.7 Оценка за курсовую работу по дисциплине выставляется по совокупности баллов по трем позициям:

- уровень реализации требований к учебному элементу курсовой работы (см. Раздел 2);
- качество подготовки сопроводительной документации;
- степень готовности проекта и его защита.

2 ТРЕБОВАНИЯ К УЧЕБНОМУ ЭЛЕМЕНТУ КУРСОВОЙ РАБОТЫ

2.1 Технические требования

Основной язык программирования для разработки программного средства в рамках курсовой работы – Python.

Обязательно наличие Git-репозитория с историей коммитов, отражающей ход работы каждого участника команды, и файлом README с описанием проекта, структуры репозитория и инструкцией по запуску.

Данные для обучения и оценки – открытые датасеты или синтетически сгенерированные данные; работа не должна предполагать доступ к реальным данным конкретной организации, если это отдельно не согласовано с руководителем.

2.2 Обязательные для реализации этапы курсовой работы

При отсутствии реализации хотя бы одного из приведённых ниже четырёх обязательных этапов – работа к защите не допускается!

1 Постановка задачи и требования. Этап включает в себя:

- формальное описание задачи (необходимо указать входные и выходные данные, ограничения);
- функциональные и нефункциональные требования к интеллектуальной системе;
- основную метрику качества с обоснованием её выбора и целевого значения, при котором результат считается приемлемым.

2 Подготовка набора данных. В рамках данного этапа необходимо:

- построить «паспорт» используемого датасета;
- выполнить разбиение на обучающую, валидационную и тестовую выборки (тестовая выборка не используется при обучении и подборе параметров);
- произвести разведочный анализ данных (EDA), включающий как минимум объём выборки, распределение целевой переменной (классов), анализ пропусков и дубликатов.

3 Модель и baseline. В рамках данного этапа необходимо выполнить:

- выбор и обоснование основной модели, которая решает задачу с помощью архитектурного подхода по теме курсовой работы (например, дообученная encoder-модель, RAG-система, CV-пайплайн);
- кроме основной модели необходимо реализовать хотя бы одно простое решение той же задачи – *baseline*, с которым сравнивается основная модель. Baseline показывает, какой результат достижим без сложного подхода, и позволяет оценить, что на самом деле даёт основная модель.

В качестве baseline можно использовать, например:

- TF-IDF и логистическую регрессию (для классификации текстов);
- правило или пороговое значение (для обнаружения аномалий);
- ту же LLM без поиска по документам (для RAG);
- предсказание самого частого класса.

4 Анализ качества результатов. Этап должен содержать:

- оценку на отложенной тестовой выборке по основной метрике и не менее чем двум дополнительным метрикам, адекватным задаче;
- сводную таблицу сравнения модели с baseline;
- анализ ошибок – матрица ошибок (для задач классификации), или разбор не менее 10 характерных ошибочных примеров с указанием их причин;
- вывод о достижении целевого значения метрики и описание ограничений полученного решения.

Для генеративных задач (RAG, чат-боты, суммаризация) оценка проводится на наборе не менее чем из 20 примеров с эталонными ответами (либо с ручной или LLM разметкой качества).

2.3 Дополнительные этапы выполнения курсовой работы

Реализация всех обязательных этапов (см. подпункт 2.2) является необходимым, но недостаточным условием для получения максимальной оценки при защите курсовой работы.

Успешное выполнение дополнительных этапов является необходимым условием для получения максимальной оценки при защите курсовой работы.

Для получения максимальной оценки при защите курсовой работы не обязательно выполнять все дополнительные этапы. Выбор дополнительных этапов также зависит от темы курсовой работы, её прикладной задачи и архитектурного подхода к решению.

Дополнительные баллы для получения максимальной оценки при защите курсовой работы зависят от качества и полноты выполнения выбранных дополнительных этапов.

1 Углублённый анализ качества входных данных. Необходимо выполнить:

- анализ качества по подгруппам данных (классам, длине входа, источникам и т. п.);
- оценку устойчивости к зашумлённым и нетипичным входным данным;
- интерпретацию решений модели (например, SHAP, LIME, карты внимания), где это применимо.

2 Сравнительный анализ моделей. Необходимо выполнить:

- сравнение не менее двух моделей или вариантов архитектурного подхода (помимо baseline);
- подбор гиперпараметров с фиксацией результатов экспериментов (например, в MLflow или в таблице экспериментов).

3 Оценка производительности решения. Необходимо выполнить:

- измерение времени отклика и пропускной способности API, потребления ресурсов (CPU/GPU, оперативная память);
- сопоставление с нефункциональными требованиями.

4 Расширенное описание хода работы. Необходимо выполнить:

- документированный код, обработки ошибок в API;
- CI-пайплайн (при возможности) (например, GitHub Actions), выполняющий тесты и сборку Docker-образа;
- фиксацию версий датасета и артефактов модели (например, DVC, MLflow Model Registry), позволяющую воспроизвести любой результат, приведённый в пояснительной записке.

5 REST API. Необходимо реализовать:

- сервис на FastAPI, содержащий как минимум эндпоинт инференса и эндпоинт проверки состояния (health check), с валидацией входных данных и автоматически сгенерированной документацией OpenAPI; API оформляется в соответствии с принципами REST, документируется (например, через автогенерируемую OpenAPI-схему FastAPI).

6 Веб-интерфейс. Необходимо реализовать:

- демонстрационный интерфейс (Streamlit/Gradio), обращающийся к системе через REST API и позволяющий выполнить основной сценарий использования;
- обработку ошибок и пустого ввода;
- отображение уверенности модели или источников ответа (для RAG).

7 Контейнеризация. Необходимо реализовать:

- Dockerfile (при нескольких сервисах – также docker-compose.yml), позволяющий запустить всю систему одной командой без ручной дополнительной настройки окружения;
- логирование входящих запросов, ответов модели и времени обработки в виде, пригодном для последующего анализа.

3 СТРУКТУРА ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ КУРСОВОЙ РАБОТЫ

3.1 Общая структура пояснительной записки курсовой работы

Пояснительная записка курсовой работы должна содержать следующие структурные элементы:

- 1 Титульный лист (см. Приложение В);
- 2 Лист с заданием (см. Приложение Б);
- 3 Реферат (Аннотация);
- 4 Содержание;
- 5 Определения, обозначения, сокращения (*при необходимости*);
- 6 Введение;
- 7 Анализ предметной области и общая постановка задачи;
- 8 Предварительная обработка и анализ данных;
- 9 Архитектурное решение;
- 10 Разработка и обучение модели;
- 11 Особенности реализации;
- 12 Анализ качества результатов;
- 13 Выводы;
- 14 Список используемых источников;
- 15 Приложение А. (обязательное) Развёртывание проекта
- 16 Приложения ... (*при необходимости*).

Необходимо строго соблюдать порядок структурных элементов пояснительной записки курсовой работы. Изменение порядка или названия разделов не допускается.

3.2 Требования к отдельным разделам пояснительной записки

Введение должно содержать объект и предмет исследования, цели и ключевые задачи разработки, актуальность разработки (рекомендуемый объём – 1–2 страницы).

Раздел 1 Анализ предметной области и общая постановка задачи должен содержать краткое описание предметной области, формальное описание решаемой задачи, обзор существующих подходов к её решению, требования к системе (функциональные и нефункциональные: производительность, масштабируемость, приватность данных).

Раздел 2 Предварительная обработка и анализ данных включает подразделы:

2.1 Сбор и подготовка данных – источники данных, «паспорт» датасета, объём выборки, статистика по признакам/классам, выявленные особенности и ограничения данных;

2.2 Разведочный анализ (EDA) – принятые решения и основные шаги по предобработке;

2.3 Углублённый анализ качества входных данных (при наличии).

Раздел 3 Архитектурное решение включает подразделы:

3.1 Описание и схема компонентов системы

3.2 Обоснование выбора конкретной модели (или, при наличии, **сравнительный анализ нескольких моделей** из числа аналогичных)

3.3 Обоснование выбора архитектурных подходов – провести сравнение с несколькими из набора, рассмотренных в курсе учебной дисциплины: classical ML, fine-tuning encoder-моделей, RAG, zero/few-shot LLM, CV-пайплайн и др.

Раздел 4 Разработка и обучение модели – описание процесса обучения (адаптации), включая использованные гиперпараметры и результаты на валидационной выборке.

Раздел 5 Особенности реализации включает подразделы:

5.1 Технологический стек (обязательный) – описание и обоснование выбора технологического стека,

а также в зависимости от реализованных дополнительных этапов выполнения курсовой работы может включать:

5.x Реализация API (при наличии) – описание эндпоинтов API,

5.x Веб-интерфейс (при наличии) – основные элементы веб-интерфейса,

5.x Контейнеризация (при наличии) – структура Docker-образа,

5.x другие сведения по реализации интеллектуальной системы (*при наличии*).

Раздел 6 Анализ качества результатов – приводятся используемые метрики и обоснование их выбора, результаты на тестовой выборке, сравнительная таблица с baseline (и другими моделями, если они рассматривались), анализ ошибок с примерами, вывод о достижении целевого

значения метрики, ограничения решения и возможные направления его улучшения.

Дополнительно (*при наличии*) указываются другие сведения об оценке качества модели.

Выводы содержат перечисление основных результатов курсовой работы, характеризующих полноту реализации задач. Результаты излагаются в форме констатации фактов, с использованием слов: «изучены», «исследованы», «разработана», «реализована», «проведена оценка», «организован мониторинг» и т. п. Текст должен быть кратким, ясным и содержать конкретные данные (рекомендуемый объём — 1–2 страницы, рекомендуемая форма – перечисление списком).

Приложение А. Развертывание проекта – является обязательным приложением к пояснительной записке, в котором прилагается ссылка на Git-репозиторий (и, при наличии реализации, описывается инструкция по развёртыванию Docker-образа).

Другие **Приложения** могут быть включены в пояснительную записку по согласованию с руководителем курсовой работы при необходимости.

4 ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ КУРСОВОЙ РАБОТЫ

Объём основной части (начиная с титульного листа и заканчивая выводом) пояснительной записки курсовой работы составляет 30–50 страниц формата А4.

Текст пояснительной записки оформляется в соответствии со стандартом предприятия СТП 01–2024 [2].

Список используемых источников должен содержать не менее 10 (десяти) источников и оформляется в точном соответствии с правилами составления библиографического списка, установленными ГОСТ 7.1-2003 [3].

Тексты программ (авторские фрагменты исходного кода, конфигурация API и Docker) при наличии обязательно приводятся в приложении(-ях) к пояснительной записке; код должен быть документирован и отражать ключевые моменты решения задачи.

Графическая часть включает схему архитектуры интеллектуальной системы (например, UML диаграмму развертывания), диаграмму данных / пайплайна, UML диаграмму вариантов использования и другие необходимые для демонстрации работы системы схемы или диаграммы. Графическая часть оформляется на листах формата А4 в соответствии с ГОСТ 19.701-90 [4], либо иным согласованным с руководителем способом для диаграмм, не являющихся классическими алгоритмическими схемами.

Текст основной части пояснительной записки (до списка использованных источников) должен быть проверен на использование заимствований с использованием одной из существующих систем. Уровень уникальности текста должен быть не менее 51%. Результат проверки указывается последним абзацем в тексте выводов по работе. К пояснительной записке прилагается отчет о проверке на заимствования.

Список используемых источников и рекомендованной литературы

[1] Положение об организации и проведении курсового проектирования в БГУИР. — Введ. 2013-12-11. — Минск: БГУИР, 2013. — 18 с.

[2] СТП 01–2017. Дипломные проекты (работы): общие требования. — Введ. 2018. — Утв. с изм. 24.11.2017. — Минск: БГУИР, 2017. — 169 с.

[3] Библиографическая запись. Библиографическое описание. Общие требования и правила составления: ГОСТ 7.1–2003. — Минск: Госстандарт Респ. Беларусь, 2004. — 48 с.

[4] ГОСТ 19.701-90 Единая система программной документации. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения.

[5] Библиографическая запись. Сокращение слов на русском языке: общие требования и правила: ГОСТ 7.0.12–2011. — М.: Стандартинформ, 2012. — 17 с.

[6] Библиографическая запись. Сокращение слов и словосочетаний на иностранных европейских языках: ГОСТ 7.11–2004 (ИСО832:1994). — Минск: Госстандарт Респ. Беларусь: Белорус. гос. ин-т стандартизации и сертификации, 2005. — 83 с.