**Особенности преподавания основ языка программирования Python студентам непрофильных специальностей**

**Цель практики:** Повысить качество обучения при большом количестве студентов (~350 ежегодно), при разгрузке курса с точки зрения насыщенности тематиками и неспособностью студентов осознавать специфические среды работы и концепции (были обнаружена неспособность большинством студентов воспринимать, осознавать и применять библиотек и, в частности, датафреймов pandas, возможностей визуализации в Python, соответствующих функций и методов).

**Описание практики:**

Перед описанием строения курса стоит упомянуть, что обучение ведется в 1 модуле 2 курса бакалавриата для 13 групп студентов в компьютерных классах на 1 академическую группу каждый. Многие студенты только в 2 семестре 1 курса бакалавриата узнают о существовании и методах работы с Excel. Языки программирования вызывают много вопросов и логического “ступора” в изучении на непрофильных специальностях нашего факультета.

Тем не менее, поскольку именно они содержат наиболее продвинутые методы анализа и визуализации, ознакомление и дальнейшая профессионализация студентов невозможна без изучения соответствующих современных языков. Без подобных дисциплин невозможна качественная реализация миссии факультета по подготовке специалистов, которые принимают решения, основанные на данных.

Таким образом, для форматирования знаний студентов и обеспечения понимания предмета при соблюдении студентами академической честности через дизайн курса, в структуре курса были применены:

* **Очные письменные элементы контроля** для проверки знаний студентов в условиях массовых списываний и использования созидающего искусственного интеллекта (Generative AI). Соответствующие элементы контроля включили в себя:

1) **Регулярные тесты (“летучки”)**, проводящиеся в начале каждого из сдвоенных семинаров на протяжении недели. 10 тестов по 5 заданий на 10 минут, включающие в себя тесты по теории, вопросы на поиск ошибки в коде, вопросы по написанию выходных данных, которые произведет код и т.д. Позволяет вспомнить теорию прошлого семинара и после проведения тестирования перейти к надстройке нового опыта на основе только что прошедшего тестирования (обеспечение “сквозного” обучения). Проводится на выдаваемых листочках, по 2 варианта на группу, задания выдаются в шахматном порядке во избежание списывания, без компьютеров;

2) **Контрольная работа**, проводящаяся в конце 1 модуля позволяет покрыть первые 6 тем, содержит 5 вариантов. Проводилась в потоковых аудиториях на 200/230 человек, что позволило использовать более глубокую рассадку студентов во избежание списывания, на выдаваемых листочках с заданием, без компьютеров;

3) **Экзамен,** аналогичный контрольной работе, но покрывающий все темы курса и проводящийся в 2 модуле.

* **Дистанционные домашние задания** были внесены для доступности студентам постоянной практики в течение курса (1 домашнее задание на 1 тему проходящую на 1 неделе). Состояло из 3 вариантов заданий, случайно назначаемых студентам через ЭИОС SmartLMS. Состояло из 2 простых, 2 средних и 1 сложной заданий внутри каждого домашнего задания.
* **Семинар на оценку**, проводимый в ЭИОС SmartLMS очно в компьютерном классе. Также имел 2 варианта заданий в 3 блоках (4 простых, 4 средних, 1 сложное), случайно назначаемых студенту. Проводилось очно в компьютерных классах на протяжении семинара.

**Цифровые инструменты**

Структура курса включила в себя следующие цифровые инструменты для обеспечения обозначенной цели:

* **​​Система SmartLMS:** хранение материалов курса, информирование студентов, прием и автоматическая проверка домашних заданий (посредством Moodle CodeRunner). Является доступным и знакомым инструментом для студентов второго курса. Не требует установки дополнительных инструментов или технических ограничений для используемых устройств.
* **Чат-бот в telegram**: система общения между студентом и преподавателем или учебным ассистентом. Представляет собой личный чат с системой тегов. Обращение студентом по конкретному тегу (личный тег преподавателя/тег по теме домашней работы/др.) определяет круг получателей сообщения. Студенты не видят обращения друг друга, что позволяет им не стесняться задавать вопросы, но при этом получать реакцию более быстро по сравнению с обращениями по электронной почте.
* **Среды работы с кодом**: преподавательским составом студентам было предложено использовать Jupyter Notebook или Google Colab. Jupyter Notebook устанавливается через дистрибутив Anaconda, который призван упростить установку Python на ПК студента, так как имеет встроенный менеджер зависимостей, способный установить язык разработки и наиболее популярные дополнительные пакеты практически без вмешательства пользователя. Google Colab является бесплатным облачным сервисом Google, с помощью которого можно выполнять программный код на Python. Такой инструмент не требует установки дополнительного ПО, его работоспособность не зависит от технических характеристик устройства, аналогично Jupyter Notebook, по умолчанию содержит популярные дополнительные модули. Студенты, которые имели опыт написания программного кода, могли выбрать любую предпочитаемую ими среду разработки, поддерживающую открытие файлов формата ipynb. Теоретический материал и задания для выполнения в течение семинара выдавались в формате ipynb — Jupyter Notebook — файл с интерактивной вычислительной средой, который позволяет выполнять код, хранить медиа и текст с разметкой Markdown.
* **Алгоритм вычисления клонов программного кода**: код, написанный на языке Python, выполняющий попарное сопоставление студенческих работ с целью выявления частичных или полных дубликатов (код-клонов). Является инструментом, помогающим преподавателю определить потенциальное наличие плагиата между студенческими работами.

Примеры описанных заданий и элементов контроля находятся в приложениях

**Образовательные результаты:**

1) Преодоление неравенства в уровне изначальной цифровой грамотности студентов (студенты прошлых годов обучения в своей массе жаловались на сложность курса https://datalens.yandex/dfl15hwwpubm1 по итогам преподавания сократилось количество комментариев про сложность курса для прохождения с точки зрения его перегрузки <https://datalens.yandex/pzi6rca8xiamb>) и введение их в изучение языков программирования, необходимых для анализа данных в рамках образовательной программы, КР и ВКР;

2) Нивелирование проблемы со списыванием студентов путем изменения дизайна элементов контроля (далее - ЭК), как следствие - повышение академической честности и соблюдение студентами этических стандартов в процессе освоения дисциплины.

### **Почему, на ваш взгляд, данная методика обеспечивает достижение поставленных целей обучения и её применение было эффективным в рамках проведённого курса?**

Письменные очные элементы контроля позволили соблюсти “сквозное” преподавание, повторение и соблюдение этических стандартов студентами в учебной деятельности, поскольку дизайн элементов контроля (бумажный, отсутствие помощников в виде бразуеров, генеративных ИИ, и средств разработки) позволяли ориентироваться исключительно на собственные знания и понимание предмета. Исключительным эксцессом явились два случайно попавших на соседние места во время контрольной работы знакомые студенты из одной академической группы (получено дисциплинарное взыскание в виде выговора), где один из студентов слово в слово переписал работу с другого. В остальном - студенты писали и старались писать работы самостоятельно, что было очень заметно на элементах контроля.

Отдельно проводимый семинар на оценку имел ту же цель - проведения элемента контроля с полным наблюдением за честностью со стороны преподавателя. Он достиг своей цели, однако, его масштабирование на элементы контроля по языкам программирования без изменения ЭИОС невозможно - внешние библиотеки, соблюдение версий и отправка/получения данных в SmartLMS имеют большое количество ограничений, что не позволяет быстро наладить постоянную работу аналогичных элементов контроля.

### **С какими трудностями вы столкнулись при реализации методики и как их преодолели?**

Летучки: для выполнения практических заданий по дисциплине необходимо проведение семинаров в компьютерных классах, что создало сложности при формировании расписания, так как компьютерных классов в корпусе гораздо меньше чем других аудиторий, более того в корпусе обучается более одного факультета, включая ШИФТ (бывш. ШФМиКН), что создало высокий спрос на комп. классы. Учитывая это и большое количество обучающихся групп, занятия по одной теме проходили на протяжении четырех дней, при этом занятий, идущих в параллель, практически не было. На каждую тему преподавателями было подготовлено два варианта летучки. Во время проведения ЭК студентам запрещалось использование электронных устройств, однако это не предотвратило утечку содержания летучек. Студенты запоминали формулировки заданий и/или ответы на них, некоторые успевали сфотографировать работу вопреки запрету. В течение первых тем курса разница между группами, писавшими в разные дни, не была замечена, однако стала более явной с течением времени – к концу недели возрастало количество идеально выполненных работ, также иногда студентами сдавались работы с ответами, идентичными правильным, но для другого варианта (вероятность того, что это произошло случайно, близка к нулевой). Более того, некоторые студенты (из групп в начале недели) открыто обсуждали тот факт, что до их коллег доходят ответы летучек (что конечно их огорчало, но не останавливало от передачи информации). Со стороны преподавателей летучки задумывались не только как способ проверить усвоение материалов предыдущей темы, но и как способ простимулировать посещаемость студентами семинаров (ЭК на бумаге невозможно написать, не присутствуя на семинаре), однако такая стратегия не оправдала ожидания, так как студенты после написания летучки (первые 10 мин. пары) массово уходили с семинаров. По этой причине организацию ЭК необходимо пересмотреть, не использовать летучки для контроля посещаемости. В качестве одного из планируемых изменений в проведении курса планируется перенести регулярное тестирование на электронную платформу. Таким образом сократятся расходы бумаги, а также времени, которое уходило на ручную проверку и выставление оценок. Освободившееся время можно перенаправить на составление большего количества вариантов летучек.

Домашние задания как элемент контроля явились основной проблемой с точки зрения академической честности. В формуле оценивания именно этот элемент контроля рассматривался студентами как наиболее простой для получения накопительной оценки, обеспечивающий им “проходной” балл по “неприоритетному”, по их мнению, курсу. Большой вес домашних заданий, их постоянность, онлайн формат выполнения задания с большим ограничением по времени (порядка недели) явился идеальным побуждающим элементом для нарушения этических стандартов со стороны студентов. Так, преподавательским составом дисциплины было обнаружено два телеграмм-канала, осуществлявших продажу домашних работ. По результатам проверки домашних работ ранее написанным скриптом на Python было обнаружена одна из действующих сетей по продаже работ, запущена служебная записка, по итогам которой 21 студент получил дисциплинарное взыскание в форме выговора. Сам дизайн элемента контроля (случайная выдача заданий из 3 вариантов) усложняет поимку списывающих, поскольку студенты в онлайн-формате могут выполнять задания, не совпадающие друг с другом но совместно, что противоречит духу элемента контроля и не позволяет преподавателям обнаружить данный факт. Тем не менее, на подобную ситуацию намекает график длительности выполнения домашних заданий в разделе оценок 8-10 за них (<https://datalens.yandex/q0k4n7blpd8kc>). В дальнейшем, по обсуждению преподавательского состава курса, было принято решение: 1) Снизить вес данного элемента контроля, 2) Провести укрупнение задания, переходя от нескольких заданий в одном к одному крупному, 3) Ввести ограничение по времени выполнения во избежание продаж решений однокурсниками.

Одной из трудностей явилась сама ЭИОС SmartLMS. Она является наиболее понятным информационным инструментом для студентов (поскольку не является чем-то исключительно новым и сложным), однако, не обладает достаточно быстрым ответом для оперативной работы и изменений условий заданий, но работает более интуитивно для сдачи заданий студентами, по сравнению с github classroom, что критично для осознанного получения знаний студентами непрофильных специальностей.

**Приложения:**

1. Пример летучки: le\_tuchka.pdf
2. Пример К/Р: midterm.pdf
3. Пример экзамена: exam.pdf
4. Пример оцениваемого семинара: graded\_sem.png
5. Пример домашнего задания: hw.png