

**Заявка на конкурс
Фонда образовательных инноваций 2022 года**

Номинация:

Оригинальные решения при проведении контрольных/оценочных мероприятий, в том числе с применением цифровых инструментов

Mentoring Challenge:

*как мы учимся программированию в команде и учимся командной работе
через программирование*

Автор:
Сенина Анна Васильевна,
преподаватель кафедры гуманитарных
дисциплин НИУ ВШЭ - Пермь
avsenina@hse.ru

1. Проблемные точки: что можно улучшить в системе контрольных мероприятий?

Курс «Основ программирования на Python» (ОП «История», НИУ ВШЭ в Перми) уже был представлен на конкурсе ФОИ (осень 2021 года), в том числе, и система контрольных мероприятий:

- учебный хакатон 0,2
- проектное предложение 0,2
- самостоятельная работа 0,3
- экзамен по программированию 0,3

В течение 2021 - 2022 уч. года курс был реализован для 75 студентов кампуса в Перми (в том числе магистрантов), частично — для 60 студентов в Санкт-Петербурге. Самостоятельная работа (решение задач) и экзамен связаны с независимым экзаменом по программированию и оставляют достаточно мало гибкости для преподавателя. Проекты хорошо показали себя: в течение 2 лет реализации курса (2021-2022, 2022-2023) студенты реализовали ряд проектов в области public history — начиная с программирования ботов и заканчивая разработкой игр на Python. Тем не менее, опыт проведения учебного хакатона позволил переосмыслить основные акценты и содержательно изменить организацию практики решения кейсов на паре командами студентов в течение ограниченного времени, а также встроить эту точку контроля в более широкий контекст посредством сквозной геймификации курса.

Основные проблемы при проведении учебного хакатона:

- ограничение в выборе способа решения кейса: **основная часть работы должна быть выполнена на Python**, решение кейса без Python приводит к понижению баллов. Если студенты не освоили базовый синтаксис Python к концу курса, им постоянно требуется помощь преподавателя;
- разный уровень владения Python у студентов (в группе в целом и в составе отдельных команд) **сложно отразить в индивидуальных оценках**;
- необходимость для преподавателя **оперативно отслеживать ход работы и прогресс** каждой команды, с учетом ограниченного времени выполнения задания.

При этом сам формат хакатона соответствует логике курса по программированию, а также в целом укладывается в основные тренды развития IT-отрасли. В качестве разработки предлагается новая система оценивания.

2. Геймификация в обучении программированию

Для курсов программирования геймификация процесса обучения становится практически нормой и требованием времени. Так, все ведущие платформы для самостоятельного обучения программированию включают в курс отдельные игровые практики (см. напр: курс [Python разработчика](#) от Яндекс.Практикум, некоторые [курсы](#) на Нетологии, курс [«Introduction to Data Science in Python»](#) на DataCamp, курсы-игры на платформе [Code Combat](#) и мн. другие).

Для курса Python также было разработано портфолио в рамках летнего курса повышения квалификации «Геймификация в онлайн-обучении» (преп. А. Сельчёнок), Приложение 1.

В систему оценивания были добавлены элементы сквозной и точечной геймификации, часть которых работают на переформатирование учебного хакатона. Выполняя задания курса в SmartLMS, в том числе, решая неочениваемые тесты для самопроверки и просматривая дополнительные материалы, студенты приобретали баллы опыта, заполняли свою шкалу прогресса на курсе.

В соответствии с набранным опытом к моменту проведения учебного хакатона, они могли продвинутся в игровой системе до уровня 4 «Ментор». Описание уровня в SmartLMS следующее: «Кажется, вы готовы обучать! Ваших знаний достаточно, чтобы взять под крыло начинающего программировать одноклассника» (Приложение 1).

В начале курса было озвучено, что студенты, вошедшие в топ-3 по игровому опыту могут пройти точку контроля — учебный хакатон — в роли ментора: наставника, консультирующего команду. Менторы получают повышенные баллы — как правило, оценки 9 и 10 в зависимости от выступления группы, которую они консультировали. Менторы также принимают участие и в оценивании других студентов. Геймификация была добровольной: студенты, не проходившие дополнительные материалы в курсе, также могли получить отличную оценку (8) при успешном выполнении задания в группе. Кроме того, наиболее сильная команда на хакатоне, выполнившая задание лучше и качественнее остальных, а также все менторы получили особые бейджи.

В игру активно включились игроки-исследователи и достигаторы, а сам хакатон для менторов стал своего рода битвой с боссом — особенно сложным заданием, за которым скрывались и особые награды: помимо самих оценок 9 и 10 это возможность особого прохождения точки контроля, требующего большего количества усилий и большей ответственности во время решения кейса командой, однако это и возможность не участвовать в итоговой презентации и внести вклад в оценку по точке контроля. Были студенты, напротив, не поддерживавшие идею геймификации, сознательно в ней не участвовавшие, что позволило им закончить курс в обычном порядке.

Таблица лидеров и загруженные в SmartLMS материалы предоставлены в Приложении 2.

3. Способ проведения учебного хакатона

В 2022-2023 году (1 модуль) учебный хакатон был проведен уже в новом формате: увеличилось количество ролей, которые студенты могли выбрать для прохождения контрольного мероприятия.

В начале пары формировались команды, каждой команде назначался ментор (добровольно, с учетом пожеланий студентов). Команда выбирала один из 5 предложенных кейсов. Каждому кейсу соответствовал датасет, а также примерный

набор гипотез (Приложение 3). Кроме того, для прохождения хакатона были разработаны общие методические рекомендации с примером организации кода в формате блокнота на Python (доступен по [ссылке](#), приложен к заявке, Приложение 4).

Команды приступали к решению кейса, ставили статистическую гипотезу, которую хотели проверить, обсуждали дизайн предстоящего мини-исследования и начинали писать код. Стоит также отметить, что предложенные кейсы 1-4 примерно схожи, однако кейс 5 отличался от остальных другой логикой решения, а также использованием специальной библиотеки `gandom`, и было озвучено на занятии, что решение кейса 5 может быть оценено выше. Тем не менее, студенты его не выбрали, ориентируясь на более знакомые по семинарским занятиям коды, которые требовалось написать для кейсов 1-4.

Менторы постоянно находились с командой: начиная с постановки гипотезы и заканчивая написанием кода, а также проверкой презентации на корректность использования базового Python. В случае проблем, которые не мог решить ментор, команды студентов могли обращаться за помощью к преподавателю, задача которого — навести на правильный ответ и дать ментору шанс затем самостоятельно вывести команду из сложной ситуации.

Менторы, кроме того, освобождают преподавателю время для контроля общего хода работ и прогресса. Преподаватель должен также следить за общим уровнем работ: важно, чтобы наиболее подготовленные студенты при условии оставшегося времени могли усилить итоговую презентацию (например, за счет использования `po-code` инструментов), а наименее подготовленные — справились с базовыми операциями в Python, которыми нужно овладеть для сдачи независимого экзамена.

Минимальный уровень владения Python для успешного решения кейса на хакатоне:

- простые типы данных `str`, `int`, `float`, `bool`
- более сложные типы данных `list`, `tuple`, `set`, `dict` (особенно `list`)
- оператор `if` и циклы `while` / `for`
- методы: `append`, `split`, `strip`, `find`, `count`
- простые функции `min`, `max`, `sum`, `len`
- сортировки: `sorted()` и `.sort()`
- файлы: синтаксис `with open`, режимы `r`, `w`, `a`

Темы курса, которые нужно повторить перед хакатоном, рассылались студентам заранее. Сами задания ориентированы на хакатоны ДН-центра НИУ ВШЭ, Яндекса и других соревнований по программированию.

Менторы заранее готовились, чтобы консультировать команды по каждой теме, остальные студенты должны были убедиться, что навыков хватает для работы в составе группы, с учетом сильных и слабых сторон участников.

Примеры заданий, а также порядок проведения хакатона отражен в презентации, Приложение 3.

4. Порядок оценивания

Оценивание курса можно разделить на два трека, в зависимости от роли на хакатоне:

Роль на хакатоне	Критерии оценивания:
Команды студентов	<p>8 баллов - решение кейса представлено в виде блокнота с кодом, сопровождается презентацией (требования к презентации - стандартные, см. гайдбук НИУ ВШЭ). Проведен анализ данных: выбрана проблема, поставлена гипотеза, гипотеза проверена с помощью программирования на Python, сделан содержательный вывод. В ходе выполнения работ продемонстрировано владение базовыми навыками программирования. В презентации обязательно отражен вклад каждого участника группы, указано распределение работ.</p> <p>6-7 баллов - допущены небольшие ошибки, неточности, код работает на проблему, однако демонстрирует пробелы в освоении необходимых тем.</p> <p>4-5 баллов - распределение работ в группе неравномерное; задачи решены без использования Python или демонстрируют слабое знакомство с программированием; проблема отсутствует, слабый анализ данных</p> <p>1-3 балла - работа не соответствует формальным требованиям (отсутствует блокнот с кодом, презентация).</p> <p>Оценка выше 8 баллов выставляется при выходе за рамки базового курса Python (например, использования библиотек Python для Data Science), кроме того, бонусный балл может быть выставлен победителям хакатона, продемонстрировавшим самую сильную работу по анализу данных (победитель определяется путем голосования).</p>
Ментор	<p>Ментор получает отличную оценку за консультирование одnogруппников</p> <p>10 баллов — группа, закрепленная за ментором, успешно справилась с решением кейса, при презентации работы не допущено ошибок в выборе и объяснении способа решения, корректно использованы основные типы данных и синтаксис Python</p>

	<p>9 баллов — группа справилась с решением кейса, однако допустила ошибки и неточности с точки зрения программирования, не повлиявшие на общий результат работы</p> <p>8 баллов — группа не справилась с решением кейса, однако ментор помогал на каждом этапе (проверяется преподавателем, который следит за общим ходом хакатона)</p>
--	---

С одной стороны, ментор сам освобожден от необходимости решать кейс, однако он несет ответственность прежде всего за корректность применения группой программирования. Так, например, группа, допустившая ошибку в презентации (обозначив результаты использования цикла `for` как сортировку, но при этом не реализовавшая использование сортировки в Python функцией `sorted` или методом `sort`), принесла ментору оценку 9, поскольку это достаточно грубая ошибка, не повлиявшая на решение кейса в целом.

Одна из задач, которые стояли передо мной для обновления этой точки контроля, - добиться более ранжированного оценивания внутри групп. Ранее группы сдавали самооценки, как правило, указывая равный вклад всех участников команды, даже в тех случаях, где в ходе хакатона большую часть задания выполнял один студент. Было решено привлечь менторов к оцениванию: за счет полного погружения в работу группы они видели распределение работ изнутри, при этом, их личная оценка не зависит от вклада других участников, следовательно, они могут принять решение самостоятельно, без согласования с группой. Кроме того, чтобы обеспечить большую честность и беспристрастность, менторы отправляют оценку преподавателю через SmartLMS, в результате чего одноклассники не знают о содержании этих оценок. Итоговое решение остается за преподавателем, с учетом рекомендаций ментора.

Другими словами, исходя из решения кейса, кода на Python и выступления с презентацией, преподаватель оценивает работу студентов, выставя базовую оценку каждой команде. Затем, за счет рекомендаций ментора, оценка отдельных участников группы может быть скорректирована в сторону повышения (но не понижения), за счет их более высокого вклада в конечный результат.

Пример рекомендаций для ментора при оценивании:

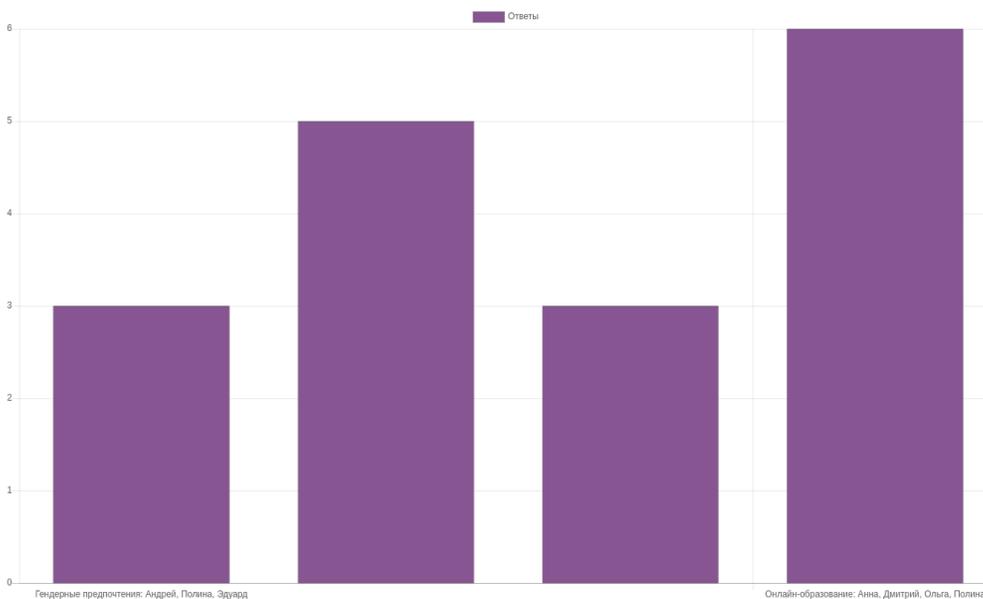
- нужно распределить 20 баллов на команду из 3 человек
- все участники не могут получить одинаковые баллы - придется ранжировать
- у оценок не может быть дробной части: только целые числа
- оценка зависит от вклада участников группы
- группа не узнает о ваших оценках
- оценки менторов будут учтены, однако строгой шкалы перевода оценки в балл нет

Идея использовать общее количество баллов, которые поровну не делятся на участников, была озвучена в одном из выступлений на последнем преподавательском марафоне победителей конкурса ФОИ, осенью 2022 года. При этом отсутствие строгой шкалы перевода оценок менторов в баллы снижает степень ответственности студентов-менторов за итоговые баллы, уровень стресса, а также позволяет избежать проблем в личных отношениях с одноклассниками после хакатона. Например, если ментор оценил работу участников как 8, 7 и 5, ему не приходится впоследствии переживать о результирующей оценке и его ответственности за «удовлетворительно», «хорошо» и «отлично» в итоговой оценке по курсу, а также об одноклассниках, заинтересованных в рейтинге, стипендии и т.д. Вместо этого, ментор присылает оценки преподавателю, и если в группе видно хорошее распределение работ, участникам, внесшим больший вклад в командную работу, преподаватель может повысить балл за хакатон. Баллы ментора на подлежат прямому переводу в баллы за курс, однако именно они позволяют выставить оценки с учетом вклада в командную работу.

Кроме того, чтобы избежать согласования оценки между ментором и группой, правила о распределении ограниченного количества баллов на группу озвучиваются в момент, когда команды заняты отправкой через SmartLMS презентации и кода на Python: так можно избежать сговора между ментором и группой и, одновременно, оставить оценки ментора неизвестными для группы. Сама возможность оценивать вклад участников для ментора озвучивается в начале хакатона, а правила распределения баллов на команду - в конце.

Помимо базовой оценки от преподавателя и оценки ментора, одна из команд объявляется победителем хакатона. Голосование было размещено в SmartLMS:

Пожалуйста, отдайте голос за лучшую работу на хакатоне (лучшую = самую интересную, понятную, полезную, технически грамотную и т. д.). Голосует каждый, можно голосовать за себя, менторы тоже голосуют



Группа победителей получила 1 бонусный балл, особый бейдж «хакатонщики», все менторы получили особый бейдж «саппорт» (см. Приложение 1 с описанием бейджей).

Итоговая формула оценивания за хакатон может быть обозначена так:

$X + M + P$, где:

X — базовый балл от преподавателя, выставленный каждой команде в зависимости от выполнения заданий хакатона (решения кейса) по критериям;

M — 0,5-1 балл, которые преподаватель может добавить, основываясь на оценках ментора;

P — бонусный 1 балл участникам команды-победителя хакатона.



Типичные примеры:

- группа вышла за пределы программы дисциплины, использовала библиотеки для анализа данных — numpy, pandas, matplotlib, seaborn, plotly и мн.др. Поскольку эти библиотеки не проходились в курсе и не освещаются в онлайн-курсе, группа и ментор получают **10 баллов**.
- группа защитилась на 8 баллов — корректно реализовала исследование, успешно выполнила все задачи по анализу данных на Python. Отдельные участники группы, выделенные ментором, получили от преподавателя +0,5/1

балл к итоговой оценке (например, баллы распределены так: участник 1 — 8 баллов, без бонуса; участник 2 — 8,5 баллов, участник 3 — 9 баллов). Если группа не стала победителем, баллы: **8, 8,5, 9**. В случае победы в хакатоне путем голосования, итоговые баллы группы увеличатся на 1: **9, 9,5, 10**. Оценка работы ментора: **10 баллов**.

- группа защитилась на 7 баллов — допущены ошибки и неточности, которые демонстрируют пробелы во владении Python, однако существенно не влияют на результат, присутствует содержательный анализ данных. Базовый балл отдельных участников группы может быть повышен (участник 1 — 7 баллов, участник 2 — 7 баллов, участник 3 — 7,5 баллов, участник 4 — 8 баллов). В случае победы в хакатоне путем голосования, итоговые баллы группы увеличатся на 1: **8, 8, 8,5, 9**. Балл ментора будет снижен за счет ошибок группы: **9 баллов**.
- ...
- Группа не подготовилась к хакатону, не справилась с заданием и не вышла на финальную защиту, однако попытки решения кейса отправлены преподавателю: участники получают по 3 балла. По оценкам ментора, балл отдельных участников группы может быть повышен до **3,5, 4 баллов**. Ментор получает **оценку 8** в случае, если помогал группе на всех этапах (решать кейс вместо группы ментор не должен).

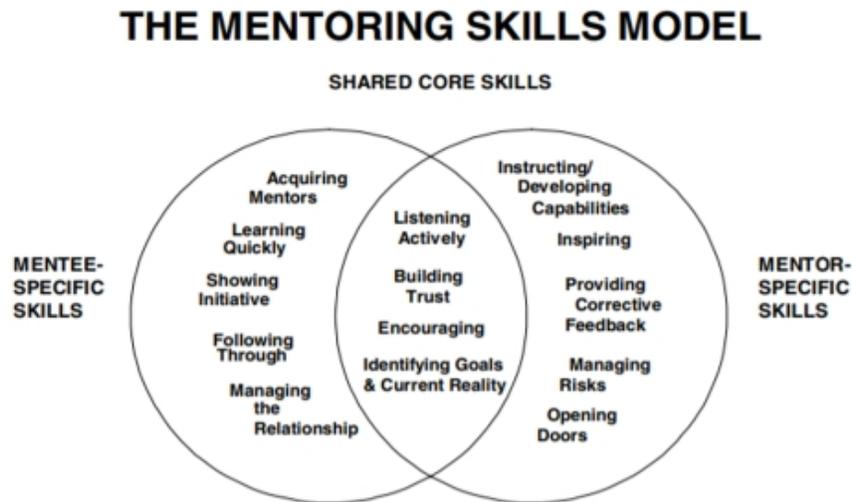
5. Результаты

Хакатон в среднем позволил поднять итоговый балл по курсу, при этом присутствует хорошее распределение итоговых оценок в группе, баллы за сам хакатон внутри малых групп также ранжированы. Точечная диаграмма показывает взаимосвязь между успешным выступлением на хакатоне и итоговой оценкой за курс. Отчет по оценкам в анонимном виде, диаграммы можно посмотреть по [ссылке](#), Приложение 5.

Менторство является обычной практикой в мире программирования: как правило, максимальный рост навыков происходит при взаимодействии с наставником, при условии разницы в навыках. Кроме того, студенты-менторы и после завершения хакатона оказывают помощь одноклассникам: они приобретают символический капитал в глазах студентов, что также положительно сказывается на освоении курса в целом. При этом, нормальной практикой в IT является не только персональная работа наставников и менти, но и фигура ментора-куратора, закрепленного за командами на хакатонах и групповых проектах. Сам хакатон и идеи его оценивания отвечают таким принципам из мира программирования, как «учись публично» (презентация работ), «обучение через вызов» (где в роли вызова выступает решение кейса за ограниченное время) и др.

В то же время, идея менторства укладывается в академические ценности НИУ ВШЭ. Так, например, студенты осваивают новые роли — ментора и менти — которые входят в [программу менторства](#) НИУ ВШЭ.

Ориентируясь на модель наставничества Л. Филлипс-Джонс, менторы и менти приобретают различные навыки, в развитии которых заинтересованы и преподаватели, помогающие их сформировать, и потенциальные работодатели:



Разработка апробирована на ОП «История», 2 курс, НИУ ВШЭ — Пермь. Она масштабируется на курсы проекта Data Culture, которыми охвачены все ОП бакалавриата, на любые другие курсы программирования.

Для **масштабирования** предложенной технологии в масштабах Высшей школы экономики (и других вузов) можно использовать систему Moodle: инструменты опыта, шкалы прогресса, бейджей и другие включены в систему или могут быть добавлены к программе дисциплины по запросу преподавателя (для других вузов). Для определения менторов используется встроенный в Moodle учет активности на курсе, и хакатону предшествует длительная работа по выполнению дополнительных заданий и просмотру размещенных преподавателем материалов. Поскольку студентам доступно их место в рейтинге в режиме онлайн, они заранее знают, кто из них выступит в роли ментора или сколько баллов активности нужно набрать, чтобы занять место в топ-3 или более участников курса и стать ментором.

Сам формат хакатона также удобно реализовывать в SmartLMS (и в целом Moodle): презентация с описанием кейсов и соответствующие им данные заранее загружаются преподавателем в систему и становятся доступными с началом пары. Процесс работы над кейсами можно реализовать через «задание» с отправкой ответа в виде текста или файла (файлы отправляют команды участников, текст со списком баллов — менторы) или «семинара» (модули, доступные преподавателям в SmartLMS). В результате вся работа выполняется через SmartLMS, где преподаватель может проверить код студентов во время их выступления или после, пока идет голосование («опрос») за победу в хакатоне, а также узнать баллы от

менторов, не сообщая их студентам-участникам хакатона. Практически все основные этапы оценивания: базовый балл за работу группы (файлы от группы) + возможные бонусы от ментора (текстовый ответ ментора) + баллы победителям хакатона (можно взять из «опроса») собраны в «задании» (или «семинаре»), что существенно облегчает процесс подсчета итоговой оценки.

В Высшей школе экономики такая реализация курса требует значительной подготовки от преподавателя для размещения и учета дополнительных материалов в течение курса, однако может быть в полном объеме быть размещена в реализации любой дисциплины в SmartLMS.

[Программировать, играя]

Анализ данных на

Python

Аудитория { бакалавры, 2 курс
магистры, 1 курс

Возраст: 19-47 лет 🤖

Гуманитарии: 95%

Айтишники: 5%

Ожидания { разные! 🤖

- "Хочу сдать курс, а то в Вышке теперь всех заставляют..."
- "Хочу попробовать это ваше программирование, (вдруг) в жизни пригодится"
- "Хочу использовать в своих исследованиях и проектах!"

Цели геймификации { общие!

- Программировать не страшно и даже весело!
- Сделаем полезный живой проект вместо (скучной) математики
- Перестанем бояться независимых экзаменов

(повторять про себя
3 раза в день!)



Подробнее о *целях*

Общие {

- Сопровождение курса: снижение уровня стресса, дополнительный материал, повторение
- Мотивация к обучению - необходима для работы с задачами онлайн-курса (требуется решить 100 задач - это пугает)
- Оценивание: задания повышенной сложности для оценок 9 и 10
- Проектный подход: создание продукта

Подробнее об

аудиторки

Бакалавры {

- ОП История -> часто с предубеждением перед "цифровыми" курсами
- За проекты и творчество, против вычислений, чисел, математики
- Их привлекает все интерактивное
- Быстро теряют мотивацию
- Боятся независимого экзамена

Магистры {

- ОП Цифровые методы в гуманитарных науках -> открыты цифровому
- Смешанный состав группы
- Разный уровень владения цифровыми компетенциями
- Требуется ранжирование заданий по уровню сложности
- Мотивация - проведение исследований



Инструмент:

бейджификация

Бейджи {



- Спасибо! Работа получена
- Задание загружено в срок



- Всадник дэдлайнов
- Все работы загружены за час до дэдлайна



- Заслужили отдых!
- Вы выполнили все домашние задания курса



- Курс окончен!
- Это было сложно, но вы справились! Все точки контроля сданы (4)



- Теперь вы волшебник!
- В программирование нет "волшебной кнопки". Но вы просто волшебник в написании инструкций! (9)



- Саппорт
- Вы успешно выступили ментором на хакатоне



- Хакатонщики
- Лучшая команда на точке контроля! (ш)кодили лучше всех





Инструмент:

бейджификация

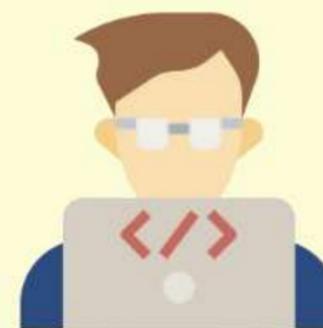
Бейджи: просмотр скрытых треков {



- Полное погружение
- Вы разблокировали скрытый модуль! Это ваш шанс копать еще глубже



- Програмируй, играя
- Вы готовы создать свою первую игру! Хотите стать геймдизайнером?



- Почти Бузмаков
- Вы действительно разбираетесь в статистике! Следующий шаг - статьи в Scopus Q1!



- Скоринкин mode on
- Вы умеете анализировать текст на Python - а между прочим Скоринкин защитил на этом диссертацию!



- Тоже digital
- Скрейпите, как на мастер-классах - теперь автоматически собрать материал для вашего исследования из интернета станет проще



Инструмент:

Опыт

Уровни {



Стажер - 0

Вы на правильном пути!

Будущий Junior - 100

Базовые навыки уже есть! Осталось накопить 10000 строк кода...



Будущий Middle - 400

Вы чувствуете себя увереннее! На нашем курсе вы уже Middle! Есть, куда расти



Ментор - 900

Кажется, вы готовы обучать! Ваших знаний достаточно, чтобы взять под крыло начинающего одногруппника



Будущий Senior - 1500

Вы достигли максимального уровня (на курсе)! Уже можно подать заявку на вакансию мечты! Мы верим, что однажды вы станете Senior-разработчиком :)



Анна Сенина avsenina@hse.ru

Инструменты

h5p, интерактивное видео

h5p {

- сценарий: пройдите первый этап трудоустройства <- Яндекс.Практикум
- далее: сценарии перед домашними заданиями

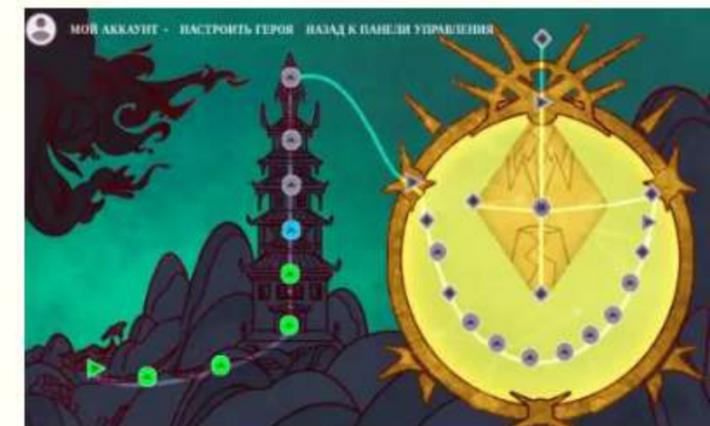
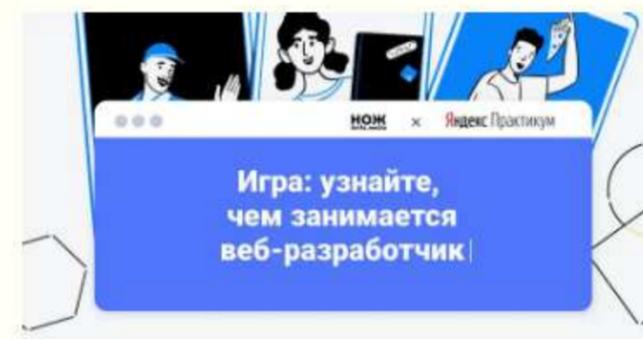
видео {

- лекции онлайн-курса
- СОП: низкие баллы онлайн-курсу
- навигация - пропуск известной / освоенной информации
- подсказки, где найти помощь по конкретным задачам онлайн-курса

Философия Python

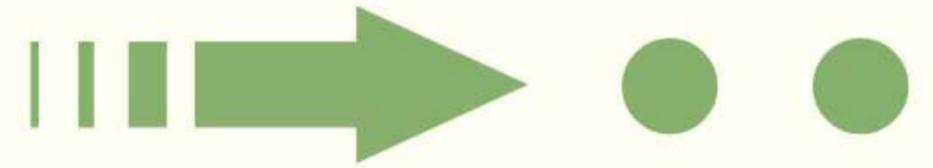


Это — Артём, и вы играете за него, как в компьютерной игре. Он менеджер по продажам, и сегодня у него не лучший день.



Анна Сенина avsenina@hse.ru

Инструменты



в5р, интерактивное видео

примеры {



Выполнить задание вы не можете. Как справимся с ситуацией?

Отправлю задание со статистикой из Excel и обязательно расскажу работу, хорошая база знаний, и программирование меня не пугает. Я готов учиться.

Попрошу друга-программиста помочь

Поздравляем с получением работы!
Осталось только научиться программированию. Начнем прямо сейчас!

Перезапустить курс

Найдите корректное название переменной

KebabCase
snake_case
CamelCase
lesson print

Что с кавычками - программа лектора сработает?

Истина
 Ложь
 Проверить

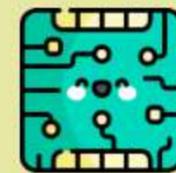


Инструмент:

тайник

Тайники {

- Печатная плата: открытые одного из скрытых треков курса без выполнения условий, на выбор студента
- Песочные часы: разовое продление дэдлайна
- Фидбек: ваша работа получит подробный отзыв
- Проекты! Открывает примеры проектов прошлых лет
- Обмен: один редкий тайник можно обменять на рекомендацию



редкий!

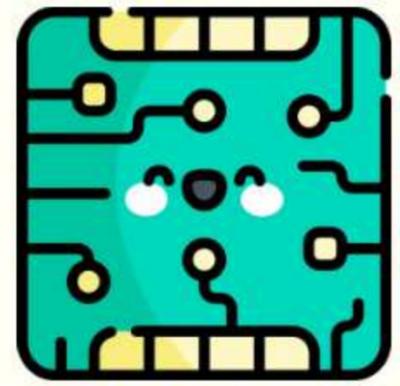


обмен



обычный

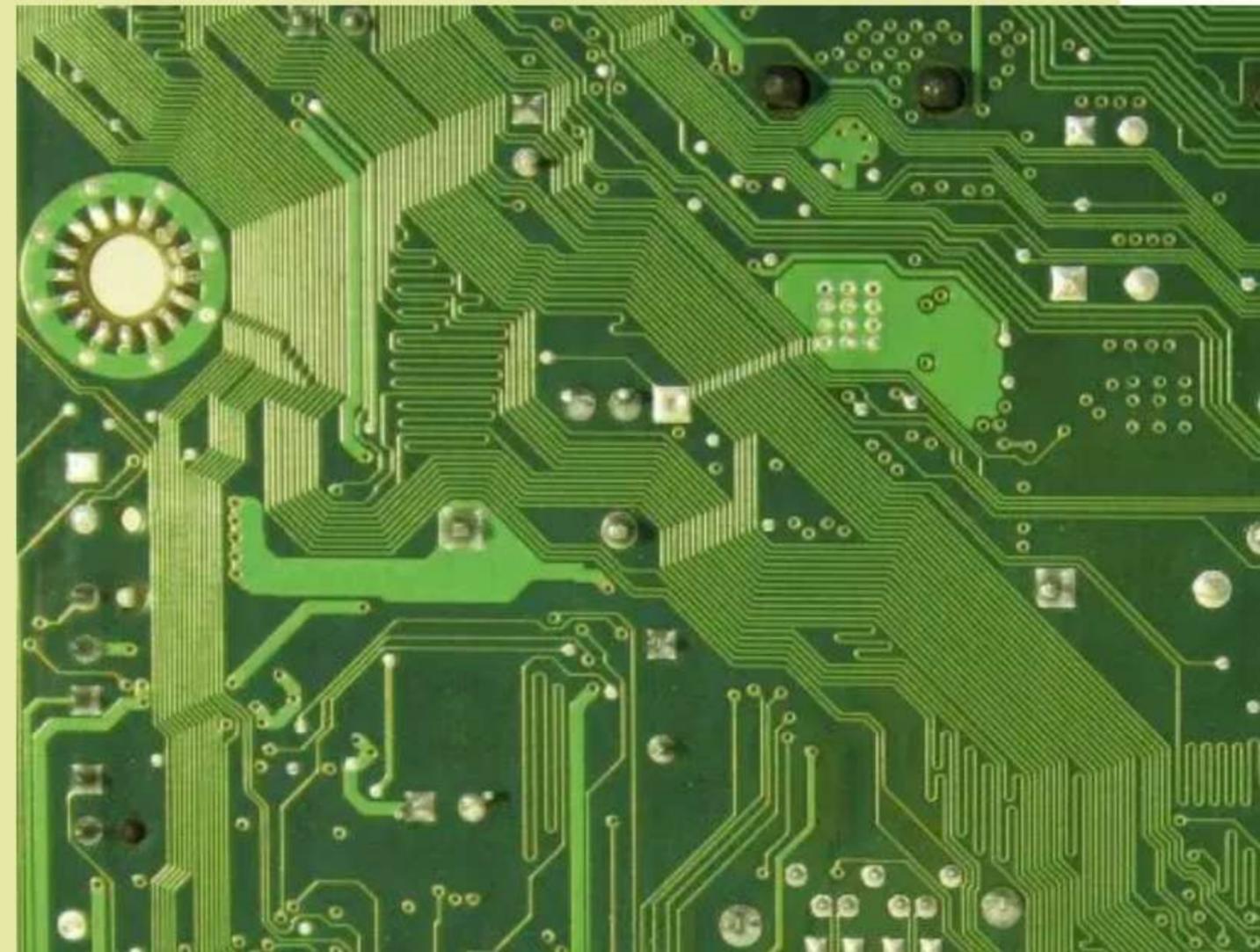
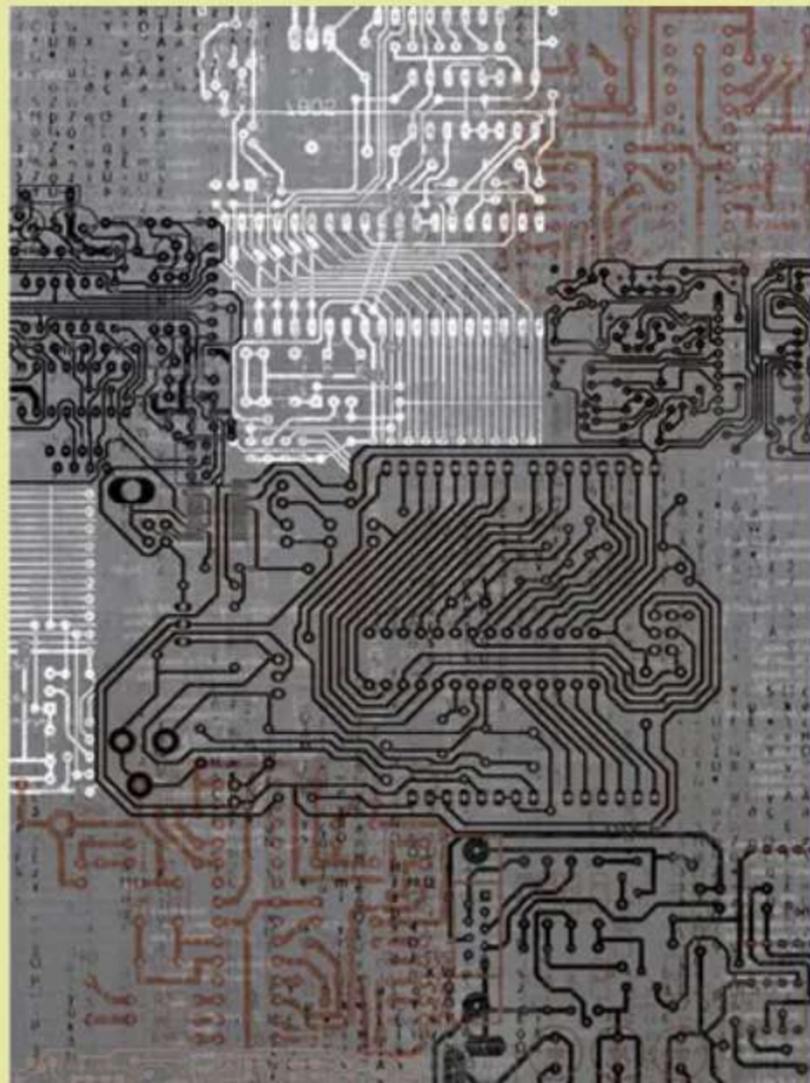
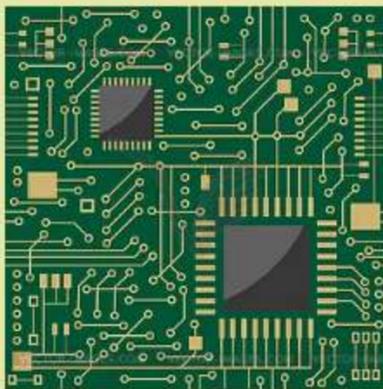
Анна Сенина avsenina@hse.ru



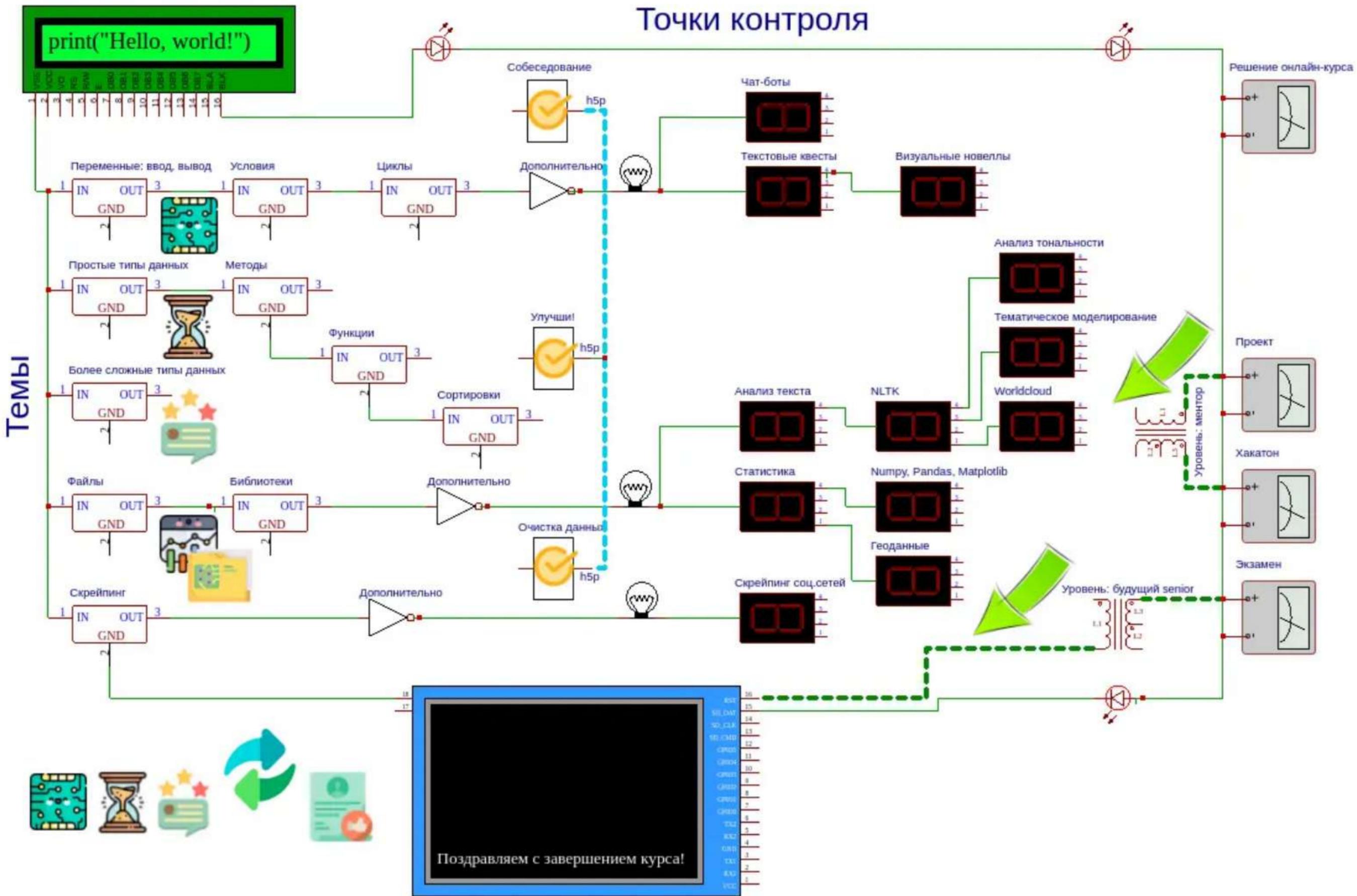
Идея для карты курса:

печатная плата

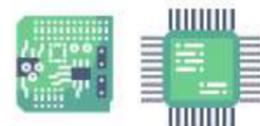
Примеры {



Точки контроля



Анна Сенина
avsenina@hse.ru



ТЕМЫ



скрытые темы



ТОЧКИ КОНТРОЛЯ

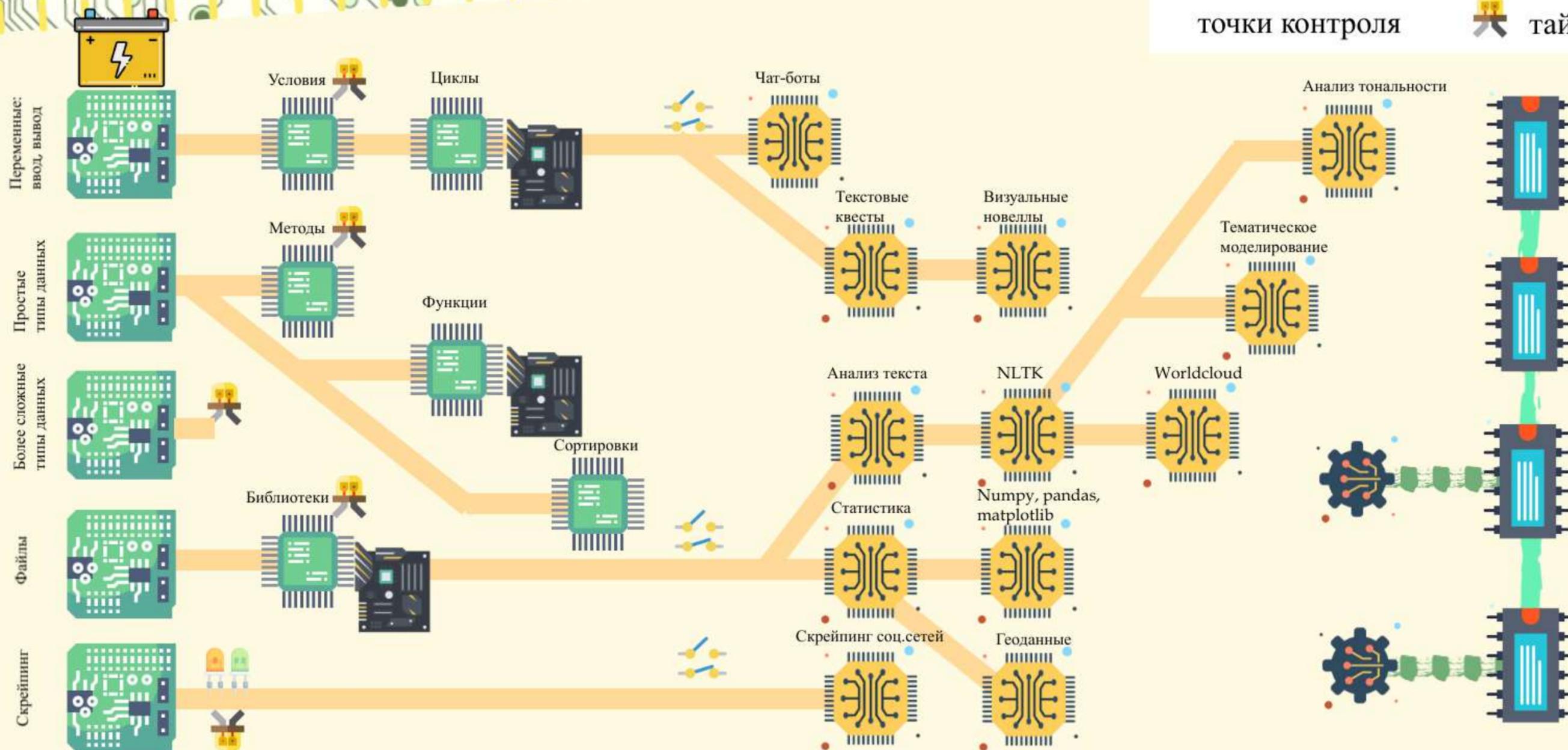


h5p



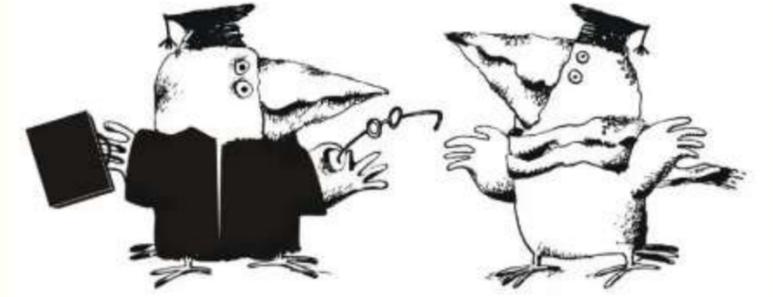
тайники

Темы



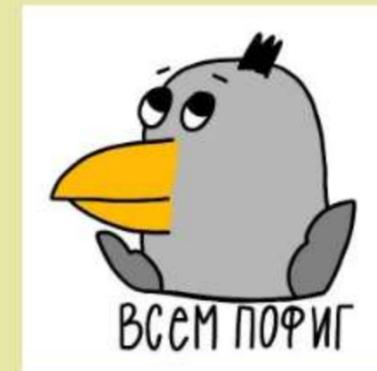
Геймификация с учетом

аудитории



Бакалавры {

- Бейджи, тайники, опыт
- Альтернативное прохождение точек контроля
- Интерактивное видео - помощь в решении обязательных задач курса
- Сценарии



Магистры {

- "Скрытые" треки
- Тайник на обмен (рекомендация)
- Опыт



Анна Сенина avsenina@hse.ru

Достижения на курсе *online.hse*

Геймификация в онлайн-обучении (1В)

Личный кабинет Личный кабинет Центр развития онлайн обучения Геймификация в онлайн-обучении (1В)

Таблица результатов

Информация Таблица результатов

Уровень	Разн.	Участник	Total	Прогресс
5	+1 570 ^{XP}	Жукова Наталья	4 005 ^{XP}	<div style="width: 100%;"><div style="width: 100%;"></div></div> 0 ^{XP} to go
5	+1 040 ^{XP}	Варламова Юлия	3 475 ^{XP}	<div style="width: 100%;"><div style="width: 100%;"></div></div> 0 ^{XP} to go
5	0 ^{XP}	Сенина Анна	2 435 ^{XP}	<div style="width: 100%;"><div style="width: 100%;"></div></div> 0 ^{XP} to go
5	-15 ^{XP}	Горелая Наталия	2 420 ^{XP}	<div style="width: 100%;"><div style="width: 100%;"></div></div> 0 ^{XP} to go
4	-530 ^{XP}	Пимонова Светлана	1 905 ^{XP}	<div style="width: 100%;"><div style="width: 95%;"></div></div> 95 ^{XP} to go
4	-750 ^{XP}	Будникова Алина	1 685 ^{XP}	<div style="width: 100%;"><div style="width: 31.5%;"></div></div> 315 ^{XP} to go

Курсы Подробности

Сенина Анна Сообщение

Подробная информация о пользователе Личный кабинет

Email
avsenina@hse.ru

Я принимаю Условия предоставления услуг

Значки
Значки из Онлайн-образование в НИУ ВШЭ:

Прекрасное выполнение домашнего задания

Тайник

Разное
Профиль полностью
Мои сертификаты
Сообщения форумов
Темы форумов
Учебные планы

Отчеты
Сегодняшние события
Все события
Краткий отчет
Полный отчет
Сессии браузера
Обзор оценок
Оценка

Входы в систему
Последний доступ к курсу
Суббота, 23 июля 2022, 19:43 (2 сек.)

Мобильное приложение

Опыт!

5

Гуру геймификации
2 435^{XP}

0^{XP} to go

Информация Таблица результатов

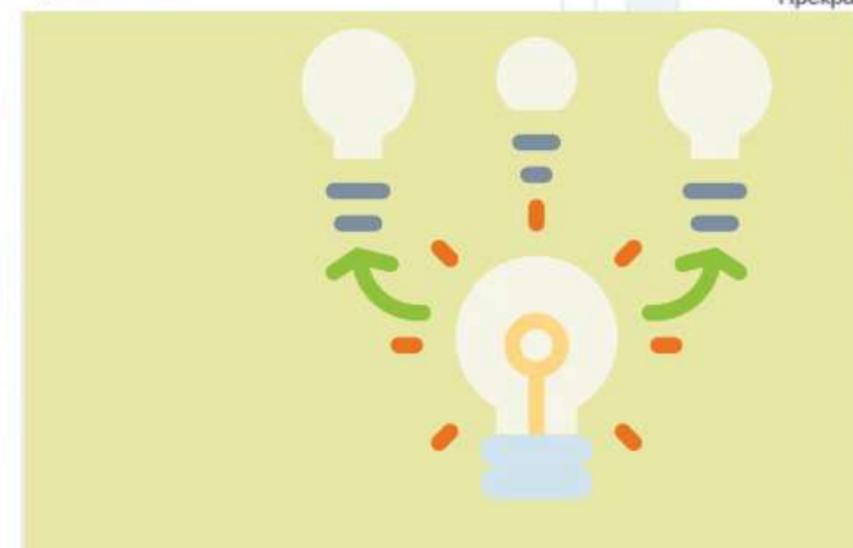
Тайник

Прогресс завершения элементов курса

Прогресс: 98%

Задание "Индивидуальное портфолио"
Не выполнено

Последние значки
 Прекрасное



Анна Сенина avsenina@hse.ru



Конкурс ФОИ 2022

игры историков

Примеры {



Основы программирования на Python (2022/2023 модули: 1)



История очная Бакалавриат 2 курс

[Личный кабинет](#)

[Курсы](#)

[Национальный исследовательский университет "Высшая школа экономики"](#)

[Филиалы](#)

[НИУ ВШЭ - Пермь](#)

[Образовательные \(учебные\) подразделения](#)

[Факультет социально-экономических и компьютерных наук](#)

[Кафедра гуманитарных дисциплин](#)

[2022](#)

[Основы программирования на Python](#)

[Основы программирования на Python \(2022/2023 модули: 1\)](#)

Таблица результатов

[Информация](#)

[Таблица результатов](#)

[Отчет](#)

[Журнал](#)

[Уровни](#)

[Правила](#)

[Внешний вид](#)

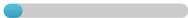
[Настройки](#)

★ [Плюс](#)

Видимые группы

Б 46.03.01 П СЭКН П 2021 очная История

Место	Уровень	Участник	Total	Прогресс
1		Конюхова Татьяна Алексеевна	2 280 ^{xp}	0 ^{xp} to go
2		Белова Александра Максимовна	1 920 ^{xp}	0 ^{xp} to go
3		Пономарева Арина Сергеевна	1 870 ^{xp}	0 ^{xp} to go
4		Гордеева Софья Олеговна	830 ^{xp}	70 ^{xp} to go
5		Сметанина Анна Сергеевна	720 ^{xp}	180 ^{xp} to go
6		Лоскутова София Игоревна	610 ^{xp}	290 ^{xp} to go
7		Басова Полина Александровна	530 ^{xp}	70 ^{xp} to go
8		Тотьмянина Светлана Михайловна	450 ^{xp}	150 ^{xp} to go
9		Герасимов Дмитрий Сергеевич	430 ^{xp}	170 ^{xp} to go
10		Шипилова Анна Игоревна	370 ^{xp}	230 ^{xp} to go
11		Терентьева Полина Владимировна	340 ^{xp}	260 ^{xp} to go
12		Яшина Ольга Дмитриевна	230 ^{xp}	70 ^{xp} to go
13		Зайцев Вадим Александрович	210 ^{xp}	90 ^{xp} to go

Место	Уровень	Участник	Total	Прогресс
14		 Харьковский Андрей Андреевич	120 ^{XP}	 180 ^{XP} to go
15		 Хисматуллова Кристина Игоревна	20 ^{XP}	 80 ^{XP} to go

[Инструкции для преподавателя](#)

[Дорожная карта по переходу на SmartLMS](#)

Техническая поддержка:

Служба поддержки для студентов

Служба поддержки для сотрудников

Данный интерфейс предназначен для студентов. Для редактирования материалов перейдите обратно в [Moodle](#).

← Основы программирования на Python (2022/2023 модули: 1)

2022

Основы программирования на Python (2022/2023 модули: 1)

Преподаватель: [Сенина Анна Васильевна](#) >

Участники:  +14 >

Оценка: [32.16668 / 40](#) >

Разделы (7)

[Скрыть все](#)



Объявления



Закрепленное



Рабочая ведомость по дисциплине

* В режиме редактирования курса перейдите в редактирование настроек элемента "Рабочая ведомость" и разместите ссылку на внешний ресурс, в котором ведете ведомость, или файл внутри SmartLMS, предварительно загрузив его.

Информация для преподавателя (скрыто от студентов):

В соответствии с [ПОПАТКУС](#) преподаватель обязан вести рабочую ведомость. Рабочая ведомость может вестись в SmartLMS. По ссылке «Рабочая ведомость преподавателя» Вы можете разместить ссылку на рабочую ведомость или файл. По умолчанию в элементе "Рабочая ведомость преподавателя" указана ссылка на данный курс, вместо нее Вам необходимо указать корректную ссылку на ведомость или файл с ведомостью.



сессии, если в расписании нет сессии (если он проводится в сессии), не позднее трех рабочих дней после определения оценок по промежуточной аттестации преподаватель направляет студентам (делает доступным для просмотра) файл с расчетом, включающим оценки по всем элементам контроля, предусмотренным программой Дисциплины.

ПУД (полная версия)

Планируемые результаты:

- Студент освоил базовый синтаксис Python и научился писать простые программы, использовать стандартные библиотеки для решения типовых задач
- Студент способен автоматизировать сбор, обработку и анализ информации из Интернета с помощью Python
- Студент умеет читать и записывать файлы, анализировать информацию в текстовой и табличной форме с помощью Python

1

Дистанционные курсы



2

Базовые элементы синтаксиса Python. Основные типы данных



Ввод-вывод. Работа с переменными. Типы данных. Условный оператор if. Цикл while. Кортежи и списки. Цикл for. Методы строк. Множества и словари. Сортировка. Функции

Тема 1. Переменные, ввод, вывод



Anaconda

Внимание! Рекомендуется использовать Anaconda (по крайней мере в компьютерном классе).

Дома можно использовать Google Colab на свой страх и риск - не независимом экзамене будут проверять на Анаконде.



 Начало курса. Вводная презентация

 Карта курса

 Тест по теме 1. Переменные, ввод-вывод. Простые типы данных 10/10

 Программирование - это космос

Темы 2-3. Условный оператор if. Цикл while

 Темы 2-3. Условия и циклы

 Условия и циклы 9.16668/10

 Условия и циклы

 Разность между датами в днях. Программа Н.А. Маткина

 Игра на Хэллоуин. Программа А. Михайловой

 Пример чат-бота нашего курса

 Python. Методы, множества, словари

 Примеры проектов

 Исходные файлы для игры Александра Костомарова

Требуется установить Ren'Py

 Исходные файлы для игры по Революции 1917 года

Требуется установить Ren'Py

 Функции: пример улучшения программы Н. Маткина (сократить, убрать повторы)

 Более сложное использование функций: генератор предложений + построение сети персонажей

3 Работа с файлами

Чтение файла. Запись в файл. Работа с текстовыми файлами. Работа с табличными данными. Анализ данных

 Блокнот 23.09 с пропусками

 Блокнот с пары

Мастер-класс по анализу данных на Python (Vyshka Digital 2022). Автор: А.

Сенина

Работа с таблицей по уровню счастья в мире, визуализация данных, построение карты, генерация открытки :)

С нуля за 1 час

 Простая статистика на примере данных ОВД-Инфо. Автор: А. Сенина

 Сложная статистика, данные о просмотре ТВ и тревожности. Задание от А.В. Бузмакова для ДН-магистров

Ссылка на датасеты: [socio](#), [form](#)

 Лемматизатор текстов (тот самый, с введения в ДН)

 Введение в NLP - простой анализ текста. Автор: Н. Маткин

Работа с текстом на Python с нуля

 Тематическое моделирование - полезный и глубокий анализ текста. Автор: Д. Скоринкин

4

Получение данных из Интернета



Веб-скейпинг. Запрос страницы. Сбор текстовых данных. Обработка таблиц сайтов. Анализ данных

 Н. Маткин. Парсер социальных сетей: ВКонтакте



статей за 2 часа)



Тезисы на конференцию (рассказ о блокноте по скрейпингу газет)

Пожалуйста, не используйте нигде этот текст и никуда его не выкладывайте - до декабря как минимум: это маленькие и не самые серьезные тезисы, но мы должны ими отчитаться по гранту, конференция состоится в декабре, сборник конференции еще не опубликован. Но! Здесь есть описание, как работает этот скрейпер, все на основе нашего BeautifulSoup, почитать полезно



Тест_3

10/10



Датасеты для хакатона



Учебный хакатон_решение



- Для групп:

Загрузите сюда финальную презентацию и ваш блокнот с кодом

Задание без блокнота не будет зачтено

- Для менторов:

загрузите оценки в виде текста (списком)



Загрузка проектных предложений



Выбор группы победителей хакатона

Пожалуйста, отдайте голос за лучшую работу на хакатоне (лучшую = самую интересную, понятную, полезную, технически грамотную и т.д.)

Голосует каждый, можно голосовать за себя, менторы тоже голосуют





6

Литература



7

Дистанционные курсы

**Техническая поддержка**

+7 (495) 772 95 90 *27817

Игнатова Наталия Геннадиевна

+7 (495) 772 95 90 *11024

Бурдюкова Елена Викторовна

Режим работы

пн.-пт. — с 9.00 до 18.00

сб.-вс. — выходной

Контактные данныеemail: lms@hse.ru

МЕНТОРЫ:
3-4 ЧЕЛОВЕКА



КОМАНДЫ
ПО 3-4 ЧЕЛОВЕКА

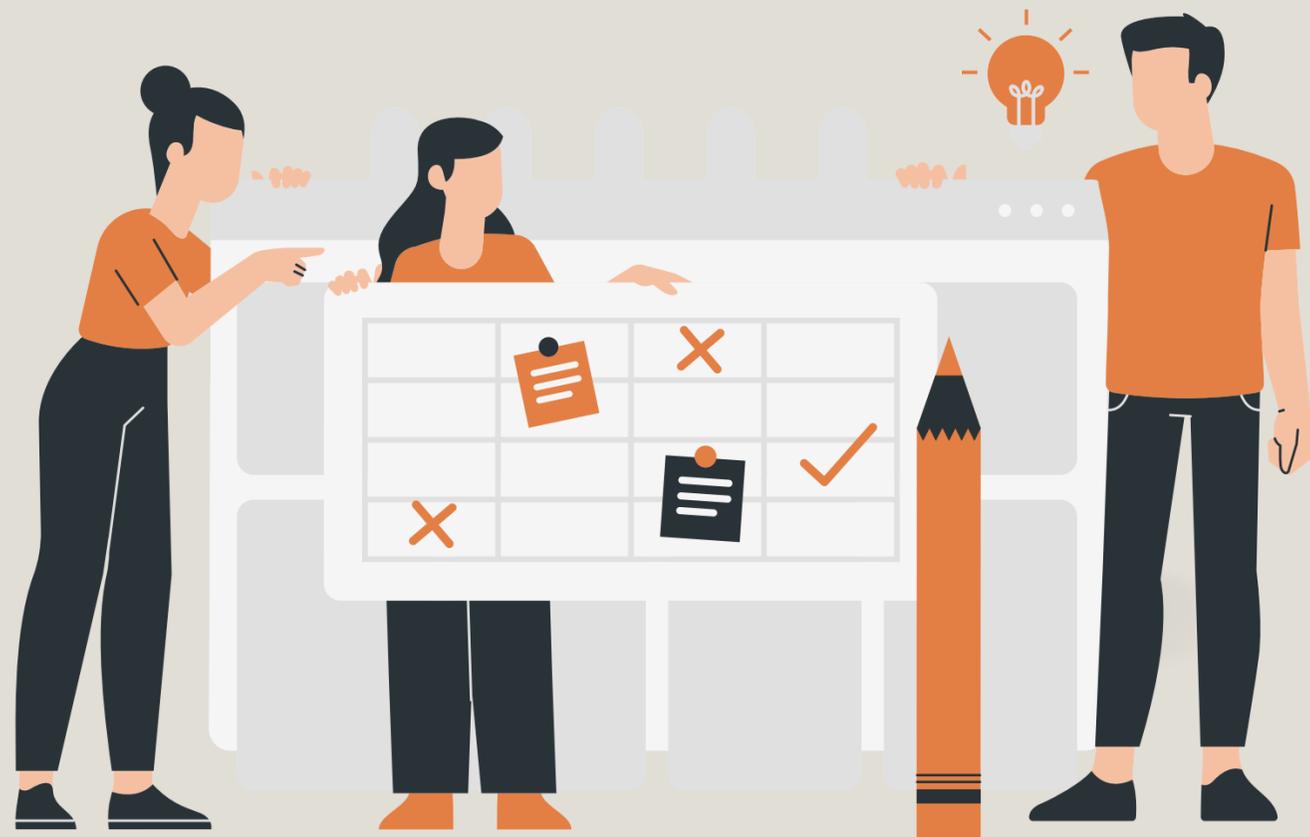


PYTHON

РЕШЕНИЕ КЕЙСОВ

УЧЕБНЫЙ ХАКАТОН

ОКТЯБРЬ 2022



1.

ВЫБРАТЬ КЕЙС

ОТКРЫТЬ ДАННЫЕ

РАЗНЫЕ ЗАДАЧИ

2.

ПРОВЕСТИ АНАЛИЗ

3.

ПОДВЕДЕНИЕ ИТОГОВ

**ВЫСТУПИТЬ
С ПРЕЗЕНТАЦИЕЙ**

СМЫСЛ ХАКАТОНА

РЕШЕНИЕ КЕЙСА





ТАБЛИЦА ЛИДЕРОВ



ЗАДАЧИ МЕНТОРА

Ментор помогает командам:

1. логика работы, код на Python
2. оценивает участников группы*

КЕЙС 1

ПСИХОЛОГИЧЕСКОЕ ЗДОРОВЬЕ СТУДЕНТОВ

ГИПОТЕЗЫ

- Проверить наличие депрессии / тревожности / наличия панических атак у студентов в зависимости от гендера или семейного положения – на выбор
- Можно поставить другой вопрос к этим данным



КЕЙС 2

ОНЛАЙН-ОБУЧЕНИЕ

ГИПОТЕЗЫ

- Проверить влияние гендера, уровня образования, финансового состояния или тип устройства (на котором проходят обучение) на уровень адаптации к онлайн-обучению – на выбор
- Можно поставить другой вопрос к этим данным



КЕЙС 3

ГЕНДЕРНЫЕ ПРЕДПОЧТЕНИЯ

ГИПОТЕЗЫ

- Составить портрет представленных гендеров: наиболее типичные предпочтения цвета, музыкального жанра, предпочитаемых напитков
- Можно поставить другой вопрос к этим данным



КЕЙС 4

НОБЕЛЕВСКИЕ ЛАУРЕАТЫ

ГИПОТЕЗЫ

- Определить наиболее типичную страну рождения нобелевских лауреатов. Для наиболее типичной страны: определить самый часто встречающийся город рождения
- ИЛИ: определить, в какой области более успешны лауреаты-женщины, а в какой – лауреаты мужчины
- Можно поставить другой вопрос к этим данным



КЕЙС 5

ГЕНЕРАТОР ХАЙКУ ✓

- Кейс сложный! Группа получит бонус в баллах
- Задача: создать генератор хайку (соединить по 3 случайных строки текста)
- Большая подсказка: нужно использовать модуль `random` – он есть в вашем файле
- Дополнительно можно попробовать создать генератор хайку на русском: для этого создайте собственную базу предложений :)





КАК ОЦЕНИВАТЬ КОМАНДУ?



ПРАВИЛА:

- 20 баллов на команду из 3 человек
- оценка – по 10-балльной шкале (все участники не могут получить 10-ки – придется ранжировать)
- оценка зависит от вклада участников группы
- группа не узнает о ваших оценках
- оценки менторов будут учтены – но строгой шкалы перевода оценки в балл нет

▼ Основы программирования на Python 2022

Это блокнот для учебного хакатона. До конца занятий обязательно загрузите этот файл в SmartLMS (1 файл от группы)

Впишите ФИО участников команды здесь:

- 1.
- 2.
- 3.

А также впишите название выбранного кейса:

▼ Для кейсов 1-4

Далее вас ждет несколько подсказок:

открытие файла:

```
with open(файл.расширение, режим, кодировка) as название_переменной(файл):  
    tab
```

Чтобы считать файл и далее работать с данными, используйте списки!

Самый удобный вариант: считать строки в файле как СПИСОК СПИСКОВ (зачем: тогда вы сможете их легко перебирать двойной индексацией)

```
for i in название_переменной(файл):  
    пустой_список.append(i.split(','))
```

Двойная индексация:

```
список[0][0] # чтобы обратиться к значению одной ячейки таблицы
```

Напоминаю также, что перебирать можно не по объектам, а по индексам:

```
for i in range(len(название_переменной)):  
    print(i)  
# напечатается 1, 2, 3.....
```

```
# пример!  
# пусть у вас есть данные о людях и их домашних питомцах
```

```
people = [['Male', 'Cat'], ['Female', 'Dog'], ['Male', 'Dog'], ['Male', 'Fish'], [
# сколько у вас всего людей?
print(len(people))
# проверьте, связан ли гендер с домашним питомцем?
m = []
f = []
for i in people:
    if i[0] == 'Male':
        m.append(i[1])
    else:
        f.append(i[1])
print(m)
print(f)

# а какие типы домашних питомцев вообще представлены?
s = set(m + f)
print(s)

# проверяем для мужчин:
print('Male')
for i in s:
    print(i, m.count(i))

# для женщин
print('Female')
for i in s:
    print(i, f.count(i))

# а можно еще и продемонстрировать владение сортировкой!
# словари неупорядоченные - почему бы снова не создать список
print('Male')
male_for_sort = []
for i in s:
    l = []
    l.append(i)
    l.append(m.count(i))
    male_for_sort.append(l)
print(sorted(male_for_sort))
# кстати, отсортировалось по алфавиту (хоть это и неочевидно для этих данных)

# давайте отсортируем по числу, например:
male_for_sort.sort(key=lambda x: x[1], reverse=True)
print(male_for_sort)
# для сортировки мы использовали в качестве ключа (= правила сортировки) безымянную
# что это такое - позже посмотрите в онлайн-курсе
# но можете попробовать на данных, не трогая все, что в аргументе key, и меняя тол
```

```
6
['Cat', 'Dog', 'Fish', 'Cat']
['Dog', 'Dog']
{'Fish', 'Cat', 'Dog'}
Male
Fish 1
Cat 2
Dog 1
Female
```

```
Fish 0
Cat 0
Dog 2
Male
[['Cat', 2], ['Dog', 1], ['Fish', 1]]
[['Cat', 2], ['Fish', 1], ['Dog', 1]]
```

▼ Для кейса 5

```
import random

people = [['Male', 'Cat'], ['Female', 'Dog'], ['Male', 'Dog'], ['Male', 'Fish'], [
print(random.choice(people))
# позволяет выбрать случайный объект из списка или другого контейнера

['Male', 'Fish']
```

▼ Ваше решение ниже:

[Платные продукты Colab](#) - [Отменить подписку](#)



