

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**  
**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**  
**Кафедра информационных технологий**

**КУРСОВАЯ РАБОТА**

**МОБИЛЬНЫЙ МОДУЛЬ ПСИХОЛОГИЧЕСКОГО ТЕСТИРОВАНИЯ**  
**ЛИЧНОСТИ**

Работу выполнил \_\_\_\_\_ М.С.Бобровский  
(подпись)

Направление подготовки 01.03.02 «Прикладная математика и информатика  
курс 3

Направленность (профиль) «Системное программирование и компьютерные  
технологии» (Математическое и программное обеспечение вычислительных  
машин)

Научный руководитель  
канд. физ.-мат. наук, доц. \_\_\_\_\_ О.В. Гаркуша  
(подпись)

Нормоконтролер  
ст. преп. \_\_\_\_\_ А.В. Харченко  
(подпись)

Краснодар  
2021

## РЕФЕРАТ

В курсовой работе 49 страниц, 13 рисунков.

МОТИВАЦИОННЫЙ ТИП ЛИЧНОСТИ ЧЕЛОВЕКА, МОБИЛЬНАЯ РАЗРАБОТКА, UNITY, C#, WEB-РАЗРАБОТКА.

Цель курсовой работы — разработка мобильного модуля психологического тестирования мотивационного профиля личности и web-страница для просмотра результатов.

При подготовке к курсовой работе изучены основные понятия построения мотивационного профиля личности, принципы разработки мобильных приложений и систем тестирования.

В практической части работы реализована методика диагностики мотивационного профиля человека (методика В.Э. Мильмана). Приложение представлено в мобильной версии, а для просмотра результатов тестирования спроектирована web-страница. Данные тестирования хранятся в векторах типа char.

## СОДЕРЖАНИЕ

Введение.....	4
1 Использование психологических тестов для определения мотивационного профиля .....	5
1.1 Общие характеристики психологических тестов определения мотивационного профиля .....	5
1.2 Описание методики В.Э. Мильмана .....	8
2 Аналогичные разработки в предметной области.....	17
3 Описание проекта.....	21
3.1 Инструментарий разработки .....	21
3.2 Алгоритм работы мобильного приложения .....	25
3.2 Алгоритм работы сайта .....	44
Заключение .....	47
Список использованных источников .....	49

## ВВЕДЕНИЕ

В современной организации результативность процесса управления персоналом и его конечных результатов зависит от множества факторов, одним из них которых является действующая система стимулирования, функционирующая в компании. Ее главная задача состоит в том, чтобы обеспечить желаемое поведение специалистов и менеджеров, необходимое для реализации долгосрочных целей компании. В свою очередь для формирования действующей, эффективной системы стимулирования следует провести исследование мотивационного профиля, который позволит выявить мотивы персонала и их (мотивов) иерархию.

Тестирование в большинстве своем занимает значительное время, поэтому для его эффективной реализации необходимо разрабатывать соответствующие инструменты. например, можно использовать мобильную версию системы компьютерного тестирования. в этом случае респондент может выполнять тест, например, по дороге на учебу. однако, хранить и анализировать данные тестирования в мобильном приложении не удобно, поэтому следует организовать связь и передачу информации в web-приложение.

Перечисленные задачи реализованы в представленной курсовой работе.

## **1 Использование психологических тестов для определения мотивационного профиля**

С появлением первых тестов наиболее употребляемым термином, с помощью которого обозначается измерение индивидуально-психологических особенностей, стал термин «психологическое тестирование». Первоначально термин «психологическое тестирование» использовался широко, включая в себя любое измерение в психологической науке. По мере развития тестов, сфера психологического тестирования сузилась до измерения личностных и когнитивных особенностей.

### **1.1 Общие характеристики психологических тестов определения мотивационного профиля**

Существует несколько определений понятия «мотивационный профиль». По мнению Р. Мартина и Ш. Ричи, мотивационный профиль личности — это индивидуальное сочетание наиболее и наименее актуальных потребностей для конкретного человека [1]. В то время как В.Э. Мильман считает, что мотивационный профиль есть совокупность устойчивых мотивов, придающих целенаправленный характер деятельности индивидов по овладению профессией, обуславливающих направленность личности на творческую самореализацию [2].

Диагностика мотивационного профиля проводится посредством сопоставления значимости ряда мотивационных факторов. Такой подход предполагает, что мотивация всегда индивидуальна, но необходимость оптимизации процессов управления, экономии времени и прочих ресурсов заставляет применять методы мотивации, ориентированные на «среднего» индивида. В основе различных типологий мотивационных профилей персонала, предлагаемых разными исследователями, как правило, лежат те или иные потребности и интересы человека. К таким потребностям могут

относиться потребность в уважении, продвижении, зарплате, стабильности, самореализации, власти, авторитете и т. д.

Мотивационный профиль представляет возможности для усовершенствования процесса отбора персонала. Он позволяет также определить не только способность человека выполнять данную работу, но и пределы его возможностей демонстрировать эффективную деятельность и способности совершенствовать свою работу. Использование мотивационного профиля в интересах отбора персонала требует, прежде всего, установления требований к желаемым профессиональным и техническим навыкам и поведению, диктуемым особенностями предлагаемого места работы.

В настоящее время существует множество методов и методик диагностики мотивационного профиля, среди которых: методика для измерения мотивации достижения (А.А. Меграбян), тест индивидуальной мотивации (лаборатория «Гуманитарные технологии»), методика диагностики источников мотивации (Дж. Барбуто и Р. Сколла), методика диагностики мотивационной структуры личности (В.Э. Мильман), методика диагностики трудовых мотивов (В.И. Герчиков), методика «Якоря карьеры» (Э. Шейн), методика диагностики мотивационного профиля личности (Ш. Ричи и П. Мартин), методика оценки мотивации достижения успеха/избегания неудач (Т. Элрес) и др.

Рассмотрим некоторые из них:

«Методика для измерения мотивации достижения» А. Меграбяна предназначена для диагностики двух обобщенных устойчивых мотивов личности: мотива стремления к успеху и мотива избегания неудачи. При этом оценивается, какой из этих двух мотивов у испытуемого доминирует.

Тест «Индивидуальной Мотивации», разработанный лабораторией «Гуманитарные технологии», позволяет выяснить основные мотивы, которыми человек руководствуется как при выборе работы и сферы деятельности, так и в процессе самой деятельности. Так же методика

позволяет построить мотивационный профиль человека, описывающий значимые и не значимые для него мотивационные факторы.

Джон Барбуто и Ричард Сколл в своем опроснике для определения источников мотивации предложили информативную модель мотивации (при этом именно в плоскости взаимоотношений «человек-организация»), интегрирующую подходы различных ученых. По мнению авторов, мотивация как явление – динамична и практически неизмерима, поэтому в модели акцент сделан на основных источниках мотивации как фундаментальных личностных образованиях.

Ш. Ричи и П. Мартин для определения индивидуального сочетания наиболее и наименее актуальных для конкретного человека потребностей, которое назвали мотивационным профилем, разработали тест. Последний дает возможность количественно оценить относительную значимость этих потребностей для конкретного человека и графически представить его мотивационный профиль. Методика «Якоря карьеры» Э. Шейна — это ценностные ориентации, социальные установки и интересы, социально обусловленные побуждения к деятельности, характерные для определённого человека. Карьерные ориентации возникают в начальные годы развития карьеры, они устойчивы и могут оставаться стабильными длительное время. При этом очень часто человек реализует их неосознанно.

Методика «Якоря карьеры» позволяет выявить следующие карьерные ориентации: профессиональная компетентность, менеджмент, автономия, стабильность, служение, вызов, интеграция стилей жизни, предпринимательство.

«Методика диагностики трудовых мотивов» В.И. Герчикова разработана специально для анализа мотивации российских работников. Она учитывает отечественную специфику и многократно отработана на десятках отечественных предприятий. Данная методика имеет ряд существенных отличий от других структурных моделей мотивации. Во-первых, она основана не на быстро меняющихся потребностях человека, а на сравнительно

устойчивых в течение трудовой жизни мотивационных типах. Во-вторых, разработана для решения управленческих, а не психологических задач. То есть она отвечает на вопрос, как добиться от работника нужного для организации поведения, а не как сделать его счастливым и удовлетворенным.

## **1.2 Описание методики В.Э. Мильмана**

В.Э. Мильман – кандидат психологических наук, старший научный сотрудник НИИ общей и педагогической психологии Академии педагогических наук СССР. Он изучал побудительные тенденции в структуре деятельности, внутреннюю и внешнюю мотивацию учебной деятельности, компоненты и уровни в функциональной структуре деятельности и разработал методику для изучения мотивационной структуры личности. С точки зрения В.Э. Мильмана, мотивационная структура личности — это совокупность мотивов, связанных и обусловленных смыслообразующим мотивом деятельности, внутренний фактор поведения, определяющий его общую направленность.

Суть методики диагностики мотивационной структуры личности заключается в выявлении некоторых устойчивых тенденций личности: общую и творческую активность, стремление к общению, обеспечению комфорта и социального статуса и др. На основе всех ответов можно составить суждение о рабочей (деловой) и общежитейской направленности личности. Методика В.Э. Мильмана позволяет определить: мотив, который является главенствующим в иерархии мотивов; насколько эти мотивы важны для личности в работе и обыденной жизни; в какой мере данные мотивы уже реализованы в жизни сейчас и в какой степени можно реализовать каждый из этих мотивов в идеале. В качестве инструментария используется опросник и бланк для ответов респондента. В данной работе мы также рассмотрим вариант использования электронных форм, которые будут представлены в виде сайта и приложения для смартфонов.



Опросник представляет собой опросный сбор диагностических симптомов по заданным мотивационным свойствам и состоит из 14 групп утверждений. Каждая из них разделяется на 8 альтернативных пунктов. Испытуемый должен выразить свое отношение к каждому из них — то есть не выбрать один из пунктов, а оценить каждый — по 5-балльной системе: да, полностью согласен (дающая 3 балла); пожалуй, да, согласен в целом (дающая 2 балла); когда как, согласен в некоторой степени (дающая 1 балла); не знаю, не задумывался над этим (дающая 0 балла); нет, не согласен (дающая 0 балла). Таким образом, испытуемый должен дать в общей сложности 112 ответов. Суммарные диагностические оценки относятся к семи мотивационным шкалам и двум шкалам эмоционального поведения — эмоциональные предпочтения и фрустрационное поведение. Включение эмоционального профиля в общий мотивационный профиль личности диктуется в научном плане общностью мотивации и эмоциональности, и в практическом — существенной диагностической информативностью.

Кроме того, для полноты диагностики общей мотивационной сферы личности каждая из семи мотивационных шкал подразделяется на 4 подшкалы:

- общежитейская (относящаяся ко всей сфере жизнедеятельности),
- рабочая (относится к сугубо рабочей или учебной сфере),
- «идеальное» состояние мотива (уровень собственно побуждения, устремления)
- «реальное» состояние — то, насколько испытуемый расценивает данный мотив удовлетворенным в настоящее время, а также то, сколько им для этого затрачивается усилий

Таким образом, общая оценка мотивационной сферы личности, по результатам тестирования, составляется из 28 шкал мотивационного профиля и 4 шкал эмоционального профиля. При необходимости разные шкалы можно

укрупнять, складывая, например, показатели шкал «общежитейская» и «рабочая», «идеальные» и «реальные» значения.

Проверочная шкала в данной методике построена наиболее простым способом: в ряде пунктов, по которым существует вероятность для испытуемого приукрасить свой портрет, есть просьба уточнить свой ответ конкретными данными. Например, если он положительно ответил на пункт о наличии хобби, то в соответствующем месте заполнения он должен уточнить, в чем именно оно состоит. Если он этого не делает, или делает неубедительно, то оценка данного ответа корректируется экспериментатором при обработке в сторону уменьшения. Ответы по этой шкале, кроме возможностей проверки и коррекции, несут в себе также дополнительную информацию об испытуемом.

В физическом варианте теста испытуемый записывает свои ответы на пункты опросника в специальном бланке. Это может происходить в индивидуальном порядке и в групповой форме, когда вопросы диктуются экспериментатором. В бланках заполнения на месте пустых клеток для ответов испытуемого даны ключи расшифровки по подшкалам. Каждая из 14 групп утверждений относится к одной из подшкал. В ряде случаев один и тот же ответ «работает на разные мотивации», относится к двум шкалам. Обычно из психометрических соображений это не принято; однако, кроме технических соображений это диктуется и содержательными моментами. В большинстве жизненных ситуаций одно и то же поведенческое проявление, как правило, полимотивировано, связано с несколькими различными мотивами; это учитывается при обработке. В результате, проявляются заметные корреляции между соответствующими шкалами, но они имеют место и в реальной жизни. При обработке результатов ответы испытуемого переводятся в баллы, баллы суммируются по каждой подшкале; в результате общая сумма может варьировать от 0 до 12 баллов.

Общая мотивационная картина личности отражается в личностно-мотивационном профиле, представляющем в количественном или графическом виде соотношение разных мотивационных шкал,

зафиксированных психодиагностически. Рассмотрим каждую шкалу поподробнее.

П — мотивация поддержания жизнеобеспечения отражает заботу субъекта о насущных условиях существования, стремление обеспечить себя и своих близких материальными ресурсами жизни. Из мотивов-интересов непосредственно включает в себя: телесные нужды, пищу, приобретение.

К — мотивы комфорта и безопасности являются, по сути, продолжением мотивации поддержания жизнеобеспечения. Это та же забота о средствах существования, но в более совершенных, удобных, надежных проявлениях, приносящих больше удовольствий. Включает в себя: комфорт, порядок, удовольствие, развлечение, приобретение, защита, безопасность.

С — статусно-престижная мотивация представляет собой мотивы поддержания жизнедеятельности и комфорта в социальной сфере. В ней отражается стремление субъекта получать внимание окружающих, престиж, положение в обществе, влияние и власть. Включает такие мотивы-интересы, как: внимание, уважение, честолюбие, доминирование.

О — мотивация общения, присоединения к группе естественна и понятна всем. В то же время может выражаться как в положительных устремлениях — аффилиация, опека, внимание, так и в социально отрицательных — унижение, угождение, подчинение, агрессия.

Д — мотивация общей активности отражает энергичность, стремление субъекта приложить свою энергию и умения в той или иной сфере деятельности. Включает в себя такие мотивы-интересы, как: активность, выносливость, упорство, возможно, противодействие.

ДР — мотивация творческой активности отражает стремление субъекта использовать свою энергию и возможности в той сфере, где можно получить творческие результаты. Включает в себя: созидание, достижение, понимание, познание.

ОД — мотивация принести общественную пользу является высшим стремлением в человеке, особенно, если при этом используются и достигаются

творческие результаты. Простой аргумент в пользу этого: если бы эта мотивация доминировала у всех или у большинства людей, то общество уверенно направлялось бы по пути к идеальному, подобно описанному А.Маслоу "эвпсихическому" сообществу.

Соотношения шкал приведенных выше, то есть характер мотивационного профиля личности, подобно отдельным мотивационным характеристикам, поддается типологизации. Рассмотрим возможные типы мотивационных профилей.

«Прогрессивный мотивационный профиль» — этот тип МП характеризуется превышением общего уровня мотивов производительного ряда (Д-ДР-ОД) над уровнем мотивов потребительного типа (П-К-С) по крайней мере на 5 баллов. Графически отражается заметным подъемом профильной линии слева направо. По полученным ранее данным этот тип положительно связан с успешностью рабочей и учебной активности. Чаще других этот тип МП встречается у людей творчески активных, в частности, у студентов-актеров, а также у школьников-пятиклассников (с более младшим возрастом этой методикой работать едва ли возможно). Одновременно, этот тип характерен и для личности с социально направленной позицией, что включается в представление о созидательной, производительной направленности личности. Была прослежена корреляция этого типа МП с типом акцентуации личности, который встречается у реально творчески ориентированных представителей художественных профессий и который А.Л. Гройсман определяет как "импрессивный". Это дополняет "прогрессивный" тип МП такими возможными поведенческими качествами как: склонность к тревожности, самонаблюдение и самокритика, недооценка своей личности, чувствительность и впечатлительность, высокий уровень притязаний и упорство в достижении выбранных целей.

«Регрессивный мотивационный профиль» — этот тип противоположен предыдущему и характеризуется превышением общего уровня потребительных мотивов над производительными, развивающими. В

наиболее отчетливом виде отражается в последовательном снижении профильной линии слева направо. При оценке этого типа следует иметь в виду, что потребительно ориентированная личность — это не просто субъект, стремящийся к удовольствиям, сибаритству, склонный к лени, эгоистичный, равнодушный к общественным интересам. Это определенный менталитет, жизненная позиция, которая в своем развитии может переходить от позиции равнодушия к социуму в позицию враждебности, вырождаясь в агрессивные, антисоциальные, деструктивные формы поведения. Этот тип можно охарактеризовать как "антитворческий", что подтверждается экспериментально.

«Экспрессивный мотивационный профиль» — этот тип МП характеризуется заметными перепадами профильной линии, в результате чего образуются пики. Чаще всего по шкалам комфорта (К) и творческой активности (ДР), хотя могут встречаться и другие сочетания, например, П и Д, С и ОД. (Критерий пика: показатель данной шкалы должен превосходить каждый из соседних на 2 или более баллов). В отличие от "прогрессивного" и "регрессивного" типов здесь общий уровень мотивов потребительного и производительного ряда существенно не отличается, но в каждом из соответствующих рядов имеются свои предпочтительные для субъекта ("пики") и отвергаемые ("завалы") шкалы. Именно это отличает данный тип МП от предыдущих не только в формальных характеристиках, но и в содержательном плане. Если в первых обращает на себя внимание направление "прогресса" или "регресса", то в данном случае — конкретные мотивационные предпочтения (и отвержения) субъекта, как бы реализующие для него единство и борьбу противоположностей; в отличие от первых двух случаев, где одна противоположность подчинена другой.

«Импульсивный мотивационный профиль» — этот тип МП отличается от "экспрессивного" большим числом пиков. Обычно их число составляет три, хотя теоретически возможно наличие и четырех. Чаще всего пики характеризуют шкалы комфорта (К), общения (О) и творческой активности

(ДР). В такой конфигурации отражается определенная конфронтация между мотивами, составляющими мотивационный профиль, так как число предпочитаемых и игнорируемых шкал в этом случае примерно равно. Это может вести к их конфликту, борьбе, общей дисгармонии мотивационной сферы в целом. Коррелирует с "импульсивным" типом акцентуации личности, к которому относятся лица инициативные, обладающие живостью характера, недостаточно сдержанные, со склонностью к риску, недостаточным контролем над эмоциями, нетерпеливостью, эмоционально неуравновешенные, недостаточно гибкие в поведенческих реакциях, снисходительные к своим слабостям. В социальных контактах могут быть обидчивыми, склонны к волнениям, колебаниям настроения, раздражительности, неуверенности и конфликтности.

«Уплощенный мотивационный профиль» — этот мотивационный профиль характерен в большей или меньшей степени сглаженным, плоским, маловыразительным рисунком, в котором отсутствуют пики, подъемы и снижения; правда, наличие одного единственного пика, чаще всего, по шкале общения не меняет характера профиля в целом. В этом типе отражается недостаточная дифференцированность мотивационной иерархии личности или ее полное отсутствие. Все основные мотивы в этой структуре практически равноценны, рядоположны для субъекта, что создает большую энтропию, неопределенность в их реализации для субъекта. При этом, если профильная линия находится достаточно высоко, это говорит о "мотивационной всеядности"— субъект любит все, хочет всего. Если профильная линия располагается низко, это говорит о мотивационном равнодушии — субъект ничего не любит, ни к чему не стремится.

С возрастом этот тип МП встречается реже. Коррелирует с типом акцентуации личности, который А.Л. Гройсман называет "сложным" (видимо, в психиатрическом плане; лиц с такой акцентуацией характеризует преобладание черт медлительности, упрямства, узости и односторонность интересов, недостаточная самостоятельность и самопонимание, они могут

быть обидчивы, подозрительны и одновременно излишне конформны. Все это, возможно, проистекает из недостаточной иерархической структурированности мотивационной сферы личности. Действительно, в теории управления известно, что плоские управляющие структуры неэффективны для работы со сложными системами.

Данный тип МП назван так, потому что обнаружил определенную корреляцию с "экспрессивным" типом акцентуации личности по А.Л. Гройсману. Представители этого типа характеризуются стремлением к самоутверждению в социуме, признании, развитым честолюбием, живостью характера, хорошим контролем над эмоциональной сферой; для них может быть характерно стремление к оригинальности, эксцентричности, лидерству, постоянному повышению уровня притязаний.

Методика широко применяется для исследования мотивационной направленности людей различных профессий. Так, например, Е.Л. Волкова в своей статье об особенностях мотивационной сферы руководителей для диагностики и изучения мотивационных аспектов личности руководителей Санкт-Петербургской таможни использовала методику В.Э. Мильмана. В ходе статистических данных была выявлена «рабочая» направленность личности руководителей, что способствует порождению общественно значимых материальных и духовных ценностей.

В статье Е.Н. Яхудиной и Ю.В. Вдовиной «Мотивационная структура и удовлетворенность трудом воспитателей ДОО» в ходе проведенного исследования было выявлено, что воспитатели с высоким уровнем общей удовлетворенности трудом (ОУТ) имеют более высокие результаты по шкале комфорт, чем воспитатели со средним уровнем удовлетворенности трудом. Полученные результаты свидетельствуют о том, что воспитатели с высоким уровнем удовлетворенности трудом будут мотивированы на комфорт, чем воспитатели со средним уровнем ОУТ [3].

Также в качестве примера использования опросника может послужить статья Р.И. Остапенко «Эмпирическое исследование особенностей

мотивационных факторов трудовой деятельности работников педагогического коллектива». Согласно результатам исследования, у 30,2% испытуемых выражен мотивационный профиль личности общежитейской направленности; 65,6% работников показали выраженность мотивационного профиля рабочей направленности .

Но, несмотря на распространённость данной методики, она имеет значительные недостатки. Обработка данных занимает много времени из-за большого количества вопросов. По этой же причине, высокая продолжительность и трудоёмкость работы утомляют респондента, что может привести к неточным ответам при тестировании и доставляет дополнительные сложности при последующей их обработке. В данной работе мы постараемся минимизировать данные проблемы путем того, что этими вычислениями будет заниматься процессор. И по итогу человеку нужно будет лишь реагировать на конечные результаты теста.



## 2 Аналогичные разработки в предметной области

Хоть и количество всевозможных тестов в интернете бескрайне огромно, но так ли много толковых среди них? Рассмотрим примеры реализации тестов именно в электронном формате. Для начала приведем пример реализации тестов, не связанных с нашей тематикой в различных приложениях.

Одним из популярных приложений для различных тестов является платформа «Testometrika», в которой включен 1131 тест. Пример одного из тестов представлен на рисунке 1.

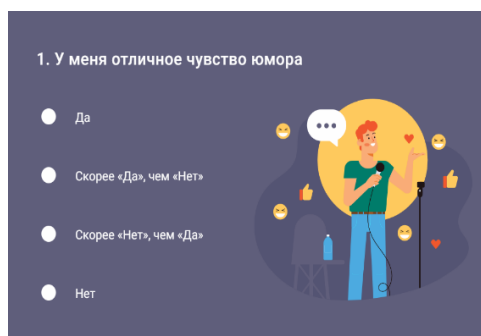


Рисунок 1 — Один из тестов «Testometrika»

Однако такое большое разнообразие тестов вызвано тем, что их создают «Testometrika team». Из-за данного подхода к созданию теста, результаты, полученные после прохождения, могут быть не действительными или не верными.

Но рассмотрим тесты на мотивационный профиль. Если выбрать только интернет страницы, то тут довольно большой выбор все возможных тестов. Например, возьмем рассмотренный нами ранее тест В.Э. Мильмана. Данный тест найден на площадке «Online Test Pad» — рисунок 2.

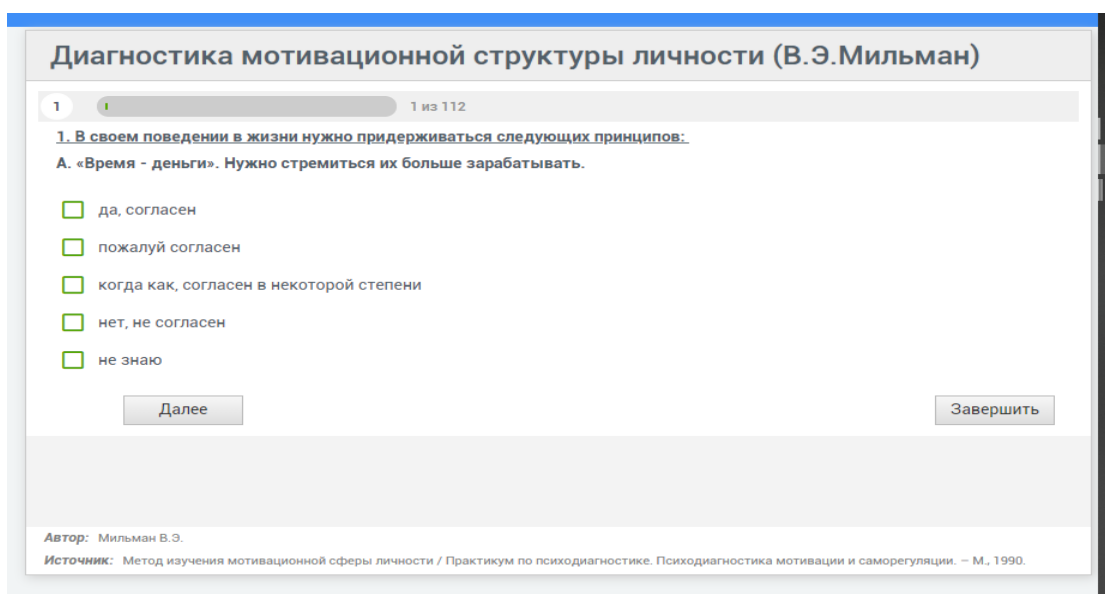


Рисунок 2 — Диагностика мотивационной структуры личности В.Э. Мильмана на площадке «Online Test Pad»

Проблемой данной реализации является не возможность выбора между версией этого теста для учащихся и работающих, которые в местах значительно отличаются. А также некоторые ответы должны подкрепляться расширенными ответами, которые в бумажной версии должны проверяться проверяющим на предмет соответствия расширенного ответа и стандартного.

Рассмотрим возможность прохождения теста на мотивационный профиль в сфере мобильных приложений. Теста на диагностики мотивационной структуры личности В.Э. Мильмана на данный момент не существует в виде приложения для мобильных или планшетов. Вообще сфера мотивационных психтестов в этом сегменте не развита. Были обнаружены лишь следующие приложения: для смартфонов на операционной системе android, это приложение «Тест на мотивацию» — рисунок 3, которое работает по не описанному тесту.

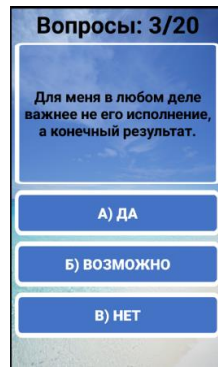


Рисунок 3 — Приложение «Тест на мотивацию»

На платформах под управлением операционной системы ios, было найдено приложение «Профиль Мотивации» — рисунок 4, которое работало по теории Герцберга, используя которую, можно разложить мотивацию человека на 8 векторов, со своими весами. Наиболее достоверный результат в данном тесте достигается при наличии ограничения по времени при оценке.

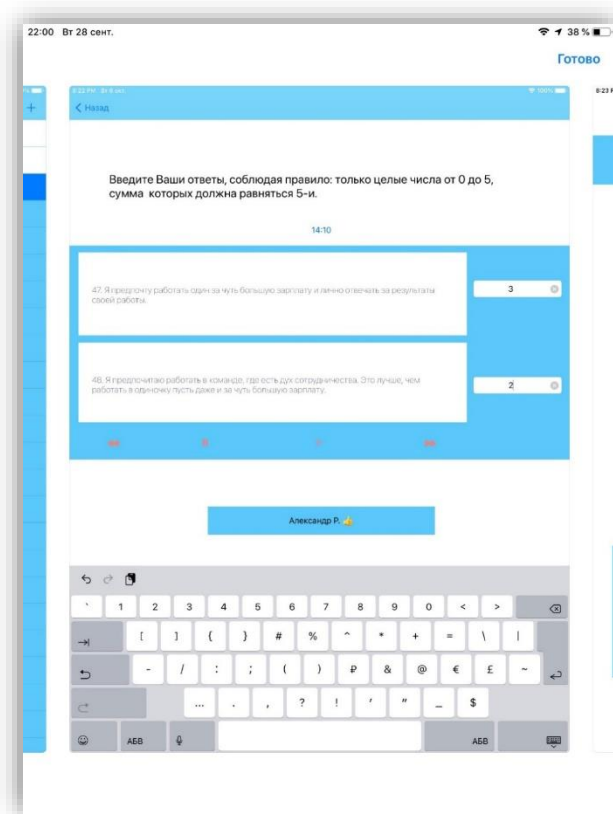


Рисунок 4 — Приложение «Профиль Мотивации»

Данное приложение является куда более серьезным инструментом в изучении своего профиля мотивации, нежели чем его предшественник, однако является платным.

Делая вывод о аналогичных приложениях и сайтах, можно понять, что онлайн психтесты достаточно популярны, но по настоящему правильно оформленных и научно обоснованных не так и много. А сфера психтестов на профиль мотивации на смартфонах представлена очень низким количеством приложений.

### 3 Описание проекта

В ходе курсовой создано приложение с психтестом В. Э. Мильмана, а именно психологический тест «Диагностики мотивационной структуры личности» В. Э. Мильмана, разработанный специально для студентов. Данный тест разработан на устройства с мобильными операционными системами ios и android, а также для обработки данных теста разработан сайт.

#### 3.1 Инструментарий разработки

IOS (ранее iPhone OS) — мобильная операционная система для смартфонов, электронных планшетов, носимых проигрывателей и некоторых других устройств, разрабатываемая и выпускаемая американской компанией Apple. Выпущена в 2007 году. Первоначально — для iPhone и iPod touch, позже — для таких устройств, как iPad. В 2014 году появилась поддержка автомобильных мультимедийных систем Apple CarPlay. В отличие от Android (Google), выпускается только для устройств, производимых фирмой Apple. В IOS используется ядро XNU, основанное на микроядре Mach и содержащее программный код, разработанный компанией Apple, а также код из ОС NeXTSTEP и FreeBSD. Ядро IOS почти идентично ядру настольной операционной системы Apple macOS (ранее называвшейся OS X). Начиная с самой первой версии, IOS работает только на планшетных компьютерах и смартфонах с процессорами архитектуры ARM [4].

Android — операционная система для смартфонов, планшетов, электронных книг, цифровых проигрывателей, наручных часов, фитнес-браслетов, игровых приставок, ноутбуков, нетбуков, смартбуков, очков Google Glass, телевизоров, проекторов и других устройств (в 2015 году появилась поддержка автомобильных развлекательных систем и бытовых роботов). Изначально разрабатывалась компанией Android, Inc., которую затем приобрела Google. Основана на ядре Linux и собственной реализации

виртуальной машины Java компании Google. Впоследствии Google инициировала создание альянса Open Handset Alliance (ОНА), который занимается поддержкой и дальнейшим развитием платформы. Android позволяет запускать Java-приложения, управляющие устройством через разработанные Google библиотеки. Android Native Development Kit позволяет портировать библиотеки и компоненты приложений, написанные на Си и других языках. В 86 % смартфонов, проданных во всём мире во втором квартале 2014 года, была установлена операционная система Android. На конференции разработчиков в мае 2017 года Google объявила, что за всю историю Android было активировано более 2 млрд Android-устройств [5].

Для реализации данного приложения выбрана среда разработки для кроссплатформенного создания видеоигр и приложений Unity. Созданные с помощью Unity игры и приложения работают в операционных системах Windows, OS X, Android, iOS, Linux, Blackberry а также на игровых приставках Wii, PlayStation 3 и Xbox 360. Приложения, созданные с помощью Unity, поддерживают DirectX и OpenGL [6].

Основной структурной единицей в Unity является “сцена”. Сцены состоят из объектов. Они могут быть использованы для создания главного меню, отдельных уровней и всего остального. Именно компоненты реализуют различные функции: рисование объектов, анимацию, физику и т.п. Такая модель позволяет собирать функциональность из простых блоков, как игрушку из конструктора Лего.

Компоненты можно писать и самому, для этого используется язык программирования C#. C# (произносится си шарп) — объектно-ориентированный язык программирования. Разработан в 1998—2001 годах группой инженеров компании Microsoft под руководством Андерса Хейлсберга и Скотта Вильтаумота [7] как язык разработки приложений для платформы Microsoft .NET Framework. C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку

операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, переменные, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML. Переняв многое от своих предшественников — языков C++, Delphi, Модула, Smalltalk и, в особенности, Java — C#, опираясь на практику их использования, исключает некоторые модели, зарекомендовавшие себя как проблематичные при разработке программных систем, например, C# в отличие от C++ не поддерживает множественное наследование классов (между тем допускается множественная реализация интерфейсов).

При создании C# скрипта внутри Unity, создаётся стандартный файл, имеющий вид:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class NewBehaviourScript : MonoBehaviour
{
    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {

    }
}
```

}

Это стандартная заготовка для компонентов. Unity подключает библиотеки `System.Collections` и `System.Collections.Generic`, а также — библиотеку `UnityEngine`, где содержится весь основной API движка. Также, Unity создает класс `NewMonoBehaviour` (название которого совпадает с именем файла). Класс унаследован от `MonoBehaviour` — это базовый класс для компонентов, создаваемых пользователем. Внутри класса Unity создает два метода: `Start` и `Update`. Эти методы движок будет вызывать сам: `Start` — сразу после того, как сцена была загружена, а `Update` — каждый кадр.

При запуске движка и создании нового проекта, появится окно, в котором можно выделить четыре основных элемента — рисунок 5.

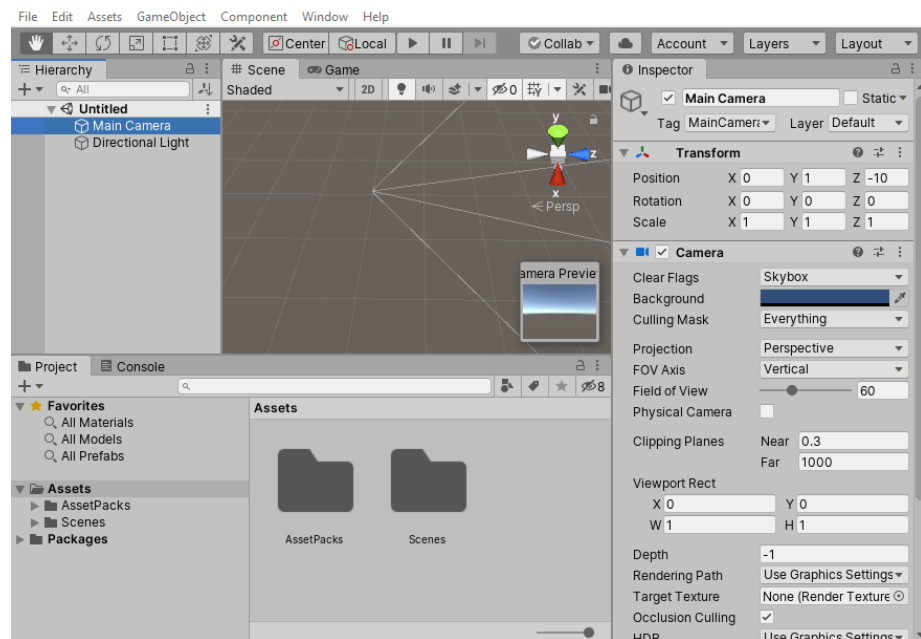


Рисунок 5 — Стандартное окно Unity

В верхнем левом углу на Рисунке 5 находится окно Иерархии (“Hierarchy”). Здесь располагается иерархия объектов в текущей открытой сцене. Unity создает два объекта при создании сцены: камеру (“Main Camera”), через которую пользователь будет видеть сцены и источник света (“Directional Light”), который будет освещать нашу сцену.



В центре находится окно редактирования сцены (“Scene”). Здесь происходит текущее отображение сцены, а также в этом разделе можно редактировать сцену визуально. Рядом находится вкладка “Game”, которая на рисунке 6 в неактивном положении. Если выбрать этот пункт, то можно будет увидеть, как сцена выглядит из объекта “Main Camera”. А если запустить проект (кнопкой со значком воспроизведения на панели инструментов), то Unity переключится на эту вкладку, где будет демонстрироваться текущая версия приложения.

В правой верхней части находится окно Инспектора (“Inspector”). В этом окне Unity показывает параметры, выбранного объекта, в этом же окне мы можем их редактировать. На рисунке 6 у камеры есть два компонента — “Transform”, который задает положение камеры на сцене и, собственно, “Camera”, который и реализует функциональность камеры.

В нижней части располагается вкладка “Project”, где можно видеть все так называемые ассеты, которые есть в нашем проекте. Ассеты — это файлы с данными, такие как текстуры, спрайты, 3d-модели, анимации, звуки и музыка, конфигурационные файлы. То есть, любые данные, которые используются для создания пользовательского интерфейса. Unity понимает большое количество стандартных форматов (например, png и jpg для картинок, или fbx для 3d-моделей), при столь большом варианте форматов проблем с загрузкой данных в проект не возникнет.

Справа от вкладки “Project” находится неактивная вкладка “Console”. В консоль Unity пишет предупреждения и сообщения об ошибках. Также в нее могут быть выведены данные для отладки приложения.

### **3.2 Алгоритм работы мобильного приложения**

Первой сценой приложения является стартовый экран где пользователь будет вводить личные данные, такие как Фамилия Имя Отчество, пол испытуемого, факультет, наименование группы учащегося, номер подгруппы

учащегося. При нажатии на кнопку старт будет исполняться процедура запускающая экран с первым вопросом теста — рисунок 6.

ФИО:

Пол:

Факультет:

Группа:

Подгруппа:

**Начать тест**

Рисунок 6 — Экрана при первом запуске приложения

При загрузке сцены выполняется C# скрипт “StartButton”, а именно функция “Start”, а при нажатии на самую кнопку запускается функция “FuncButton”.

```
using System.Collections;  
using System.Collections.Generic;  
using UnityEngine;  
using UnityEngine.SceneManagement;
```

```

using UnityEngine.UI;

public class StartButton : MonoBehaviour
{
    public InputField InputFullName;
    public InputField InputGender;
    public InputField InputFaculty;
    public InputField InputClass;
    public InputField InputSubClass;
    public GameObject ErrorText;

    void Start(){
        if(PlayerPrefs.HasKey("currentSceneMilman")){

SceneManager.LoadScene(PlayerPrefs.GetInt("currentSceneMilman"));

        }
    }

    public void FuncButton(){
        if(InputFullName.text == "" || InputGender.text == ""
|| InputFaculty.text == "" || InputClass.text == "" ||
InputSubClass.text == ""){
            ErrorText.SetActive(true);
        }else{
            PlayerPrefs.SetString("FullName",
InputFullName.text);
            PlayerPrefs.SetString("Gender",
InputGender.text);

```

```

        PlayerPrefs.SetString("Faculty",
InputFaculty.text);
        PlayerPrefs.SetString("Class", InputClass.text);
        PlayerPrefs.SetString("SubClass",
InputSubClass.text);
        PlayerPrefs.SetInt("currentSceneMilman",0);
        int currentScene =
PlayerPrefs.GetInt("currentSceneMilman") + 1;

PlayerPrefs.SetInt("currentSceneMilman",currentScene);

SceneManager.LoadScene(PlayerPrefs.GetInt("currentSceneMilman"));
    }
}
}

```

Функция “Start” проверяет работал ли пользователь в программе ранее и загружает последнюю сцену на которой остановился пользователь, если таковая имеется. Функция “FuncButton”, которая вначале проверяет поля на пустоту и, если все поля заполнены, запоминает их в PlayerPrefs. PlayerPrefs - это класс, в котором хранятся данные пользователя и сохраняются между сценами. Данные этого класса могут записываться и перезаписываться через “ключ”, путем статических методов GetString и SetString для данных типа String, SetFloat и GetFloat для данных типа Float, SetInt и GetInt для данных типа Int. В данном примере мы сохраняем значения полного имени, факультета и курса в переменные с ключами “FullName”, “Gender”, “Faculty”, “Class” и “SubClass” соответственно. После чего программа инициализирует

все нужные для теста переменные, после чего при помощи класса SceneManager и метода LoadScene загружается сцена с первым вопросом “Question 1” — рисунок 7.

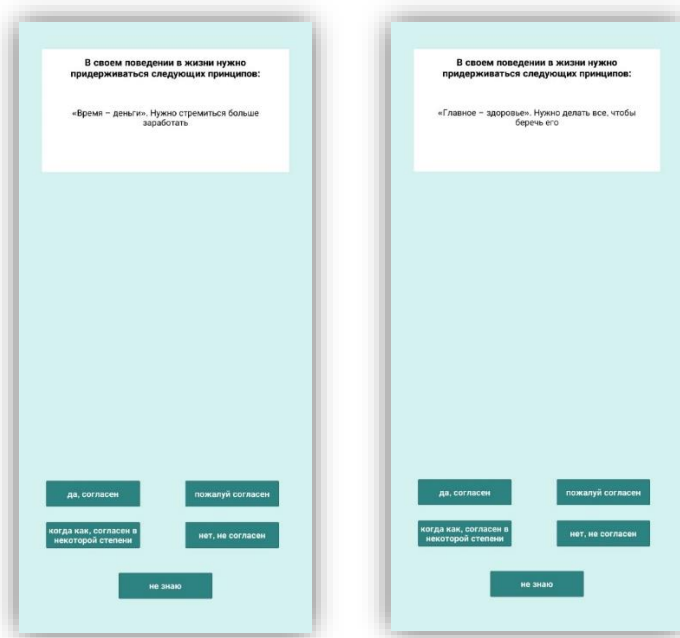


Рисунок 7 — Экраны с вопросами

В данных сценах используются пять кнопок с однотипными ответами: да, согласен (дающая 3 балла); пожалуй, согласен (дающая 2 балла); когда как, согласен в некоторой степени (дающая 1 балла); не знаю (дающая 0 балла); нет, не согласен (дающая 0 балла). Баллы будут суммироваться для каждой отдельной категории: П, К, С, О, Д, ДР, ОД, которые были рассмотрены в пункте 1.2. Также мотивационные профили делятся на 2 вида: общежитейская, рабочая, а также по состоянию мотива на «идеальное» и на «реальное», которые были рассмотрены в пункте 1.2. При нажатии любой кнопки будет запускаться скрипт соответствующий категории вопроса. Так, например, в первом вопросе добавляются характеристики типа П (мотивация поддержания жизнеобеспечения).

```
using System.Collections;
```

```

using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;
public class StandartButton : MonoBehaviour
{
    [SerializeField] private int Score;
    [SerializeField] private string[] characteristic;

    public void FuncButton(){
        for(int j = 0; j<characteristic.Length; j++){
            if(PlayerPrefs.HasKey(characteristic[j])){
                int i =
PlayerPrefs.GetInt(characteristic[j]);
                i+=Score;
                PlayerPrefs.SetInt(characteristic[j], i);
            }
        }
        int currentScene =
PlayerPrefs.GetInt("currentSceneMilman") + 1;

PlayerPrefs.SetInt("currentSceneMilman",currentScene);

SceneManager.LoadScene(PlayerPrefs.GetInt("currentSceneMilma
n"));
    }
}

```

При нажатии на кнопку выполняется скрипт “StandartButton”, а именно функция “FuncButton”. Данный скрипт является универсальным для

практически всех вопросов внутри теста, кроме вопросов с уточняющими данными, которые будут рассмотрены чуть позднее. Универсальность обоснована переменными “NextScene”, ”Score” и “characteristic”, которые будут записаны внутри каждой сцены заранее. Функция “FuncButton” проверяет существует ли класс PlayerPrefs с заданной характеристикой, через метод “HasKey()”, который возвращает значения типа true или false, соответственно. После чего если получаем значение true записываем предыдущие значение в переменную i, прибавляем значение баллов соответствующее кнопке (переменная Score) и перезаписываем значение при помощи метода SetInt. После чего загружаем следующую сцену через класс SceneManager, метод LoadScene и переменную означающую позицию сцены “NextScene”.

Также в приложении присутствуют сцены, в которых нужно вводить уточняющие данные — рисунок 8.



Рисунок 8 — Экран с вводом уточняющих данных

В некоторых экранах присутствует проверка ответов на вопрос путем уточнения ответа. Например, на вопрос “занимает ли ваше свободное время

общественные дела?” При утвердительном ответе пользователю, нужно будет указать каким именно общественным делом занимается пользователь. Если же пользователь укажет утвердительный ответ и при этом не укажет каким именно, баллы за этот ответ будут занижены и будет добавлены баллы в шкалу недостоверной информации.

В приложении это действие реализовано при помощи отдельного скрипта, который немного отличается от стандартного для кнопок.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using UnityEngine.SceneManagement;

public class StandartButtonCheckLies : MonoBehaviour
{
    public InputField InputClarifyingInformation;
    [SerializeField] private int Score;
    [SerializeField] private string[] characteristic;

    public void CheckLies(){
        if(InputClarifyingInformation.text == "" && Score >
0){
            if(Score == 3){
                Score=1;
            }else{
                Score=0;
            }
        }
    }
}
```



```

        for(int j = 0; j<characteristic.Length; j++){
            if(PlayerPrefs.HasKey(characteristic[j])){
                int i =
PlayerPrefs.GetInt(characteristic[j]);
                i+=Score;
                PlayerPrefs.SetInt(characteristic[j], i);
            }else{
                PlayerPrefs.SetInt(characteristic[j],
Score);
            }
        }
        int currentScene =
PlayerPrefs.GetInt("currentSceneMilman") + 1;

PlayerPrefs.SetInt("currentSceneMilman",currentScene);

SceneManager.LoadScene(PlayerPrefs.GetInt("currentSceneMilma
n"));
    }
}

```

Этот скрипт отличается от предыдущего добавлением проверки поля для уточняющей информации на пустоту. И если количество баллов за ответ больше нуля и дополнительное поле пустое, скрипт занизит ответ. После чего происходят те же самые операции что и в прошлый раз.

После 112 вопросов приложение будет выводить результат, который можно увидеть на рисунке 9.

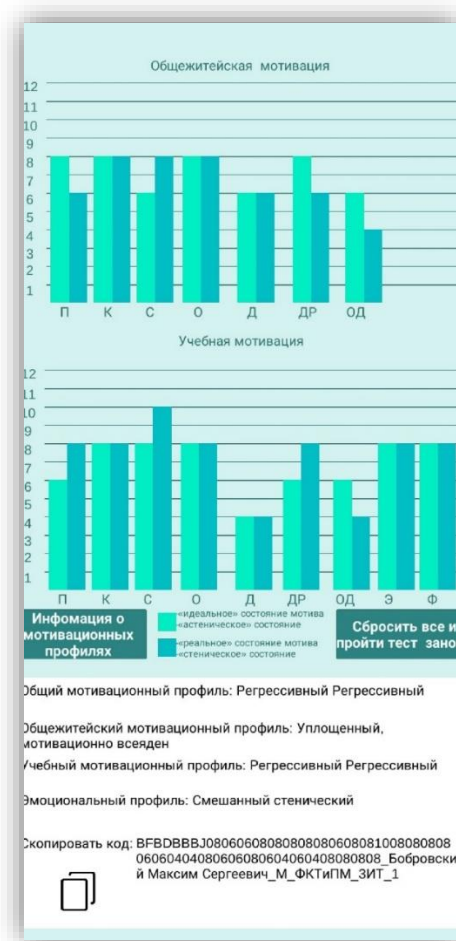


Рисунок 9 — Экран с результатами

Итоговая сцена формируется при помощи скрипта “ItogRezult”, который инициализируется при загрузке сцены и запускает функцию “Start()”.

```
for(int i=0; i<columnOfCharacteristic.Length;i++){
    if(PlayerPrefs.HasKey(characteristic[i])){

        columnOfCharacteristic[i].transform.localScale
        = new Vector3(0.2f, 0.2f *
        PlayerPrefs.GetInt(characteristic[i]),
        0.0f);
        columnOfCharacteristic[i].transform.position
        += new
```

```

        Vector3(0.0f, (PlayerPrefs.GetInt(character
            istic[i]) * 0.2f)/2 - 0.1f, 0.0f);
    }else{
        Debug.Log(characteristic[i]);
    }
}

```

Выше приведен фрагмент кода из функции “Start()”, который выводит диаграмму ответов, путем увеличения высоты каждой из колон на размер соответствующий ответу.

Для описания последующих действий в функции “Start()”, необходимо ввести понятия следующих вспомогательных функций: “GetallCharacteristic()”, “CheckProgressiveRegressiv()”, “CheckImpulsiveExtrovertedFlattend()”.

Функция “GetallCharacteristic()” преобразует массив из отдельных характеристик в массив с конкретными, интересующими нас характеристиками.

```

float[] GetallCharacteristic(int divisor,int choice){
    float Dcharacteristic =0;
    for(int i=0; i<divisor; i++){
        Dcharacteristic +=
PlayerPrefs.GetInt(characteristic[16+i + choice]);
    }
    Dcharacteristic = Dcharacteristic/divisor;
    float DRcharacteristic =0;
    for(int i=0; i<divisor; i++){
        DRcharacteristic +=
PlayerPrefs.GetInt(characteristic[20+i+ choice]);

```

```

    }
    DRcharacteristic = DRcharacteristic/devisor;
    float ODcharacteristic =0;
    for(int i=0; i<devisor; i++){
        ODcharacteristic +=
PlayerPrefs.GetInt(characteristic[24+i+ choice]);
    }
    ODcharacteristic = ODcharacteristic/devisor;
    float Ocharacteristic =0;
    for(int i=0; i<devisor; i++){
        Ocharacteristic +=
PlayerPrefs.GetInt(characteristic[12+i+choice]);
    }
    Ocharacteristic = Ocharacteristic/devisor;
    float Ccharacteristic =0;
    for(int i=0; i<devisor; i++){
        Ccharacteristic +=
PlayerPrefs.GetInt(characteristic[8+i+choice]);
    }
    Ccharacteristic = Ccharacteristic/devisor;
    float Kcharacteristic =0;
    for(int i=0; i<devisor; i++){
        Kcharacteristic +=
PlayerPrefs.GetInt(characteristic[4+i+choice]);
    }
    Kcharacteristic = Kcharacteristic/devisor;
    float Pcharacteristic =0;
    for(int i=0; i<devisor; i++){

```

```

        Pcharacteristic +=
PlayerPrefs.GetInt(characteristic[0+i+choice]);
    }
    Pcharacteristic = Pcharacteristic/devisor;
    float[] allCharacteristic = {Pcharacteristic,
Kcharacteristic, Ccharacteristic, Ocharacteristic,
Dcharacteristic, DRcharacteristic, ODcharacteristic};
    return allCharacteristic;

```

Функция “CheckProgressivRegressiv()” получает на вход массив “allCharacteristic” из функции “GetallCharacteristic()”, а также текстовое поле “Rezult”, куда будет выводиться ответ на экране. На основании данных из массива “allCharacteristic”, определяется является ли пользователь человеком с “Прогрессивным” мотивационным профилем или же с “Регрессивным” мотивационным профилем. Эта функция выводит результат на экран в заданное поле “Rezult”, а также возвращает букву, которая будет записана в уникальный код для каждого пользователя. Если же пользователь не обладает ни “Прогрессивным” или “Регрессивным” мотивационным профилем, функция не взаимодействует с пользователем, при таком развитии сценария мотивационный профиль будет определен в следующей функции “CheckImpulsiveExtrovertedFlattened()”.

```

string CheckProgressivRegressiv(float[] allCharacteristic,
Text Rezult){
    if((allCharacteristic[4] + allCharacteristic[5] +
allCharacteristic[6])-(allCharacteristic[0] +
allCharacteristic[1] + allCharacteristic[2]) >= 5){
        Rezult.text += " Прогрессивный";
        return "A";
    }else{

```

```

        if((allCharacteristic[0] + allCharacteristic[1]
+ allCharacteristic[2])-(allCharacteristic[4] +
allCharacteristic[5] + allCharacteristic[6]) >= 5){
            Rezult.text += " Регрессивный";
            return "B";
        }else{
            return "";
        }
    }
}
}

```

Функция “CheckImpulsiveExtrovertedFlattened()” также как и предыдущая функция получает на вход массив “allCharacteristic” из функции “GetallCharacteristic()”, а также текстовое поле “Rezult”, куда будет выводиться ответ на экране. На основании данных из массива “allCharacteristic”, определяется является ли пользователь человеком с “Импульсивным”, “Экспрессивным”, “Уплощенным, мотивационно всеядным” и “Уплощенным, мотивационно равнодушным” мотивационным профилем. Эта функция выводит результат на экран в заданное поле “ Rezult”, а также возвращает букву, которая будет записана в уникальный код для каждого пользователя.

```

string CheckImpulsiveExtrovertedFLattened(float[]
allCharacteristic, Text Rezult){
    if(CheckProgressivRegressiv(allCharacteristic,
Rezult) != ""){
        return "";
    }else{
        int pick=0;
        for(int i =0; i<allCharacteristic.Length; i++){

```

```

        switch(i){
            case 0:
                if(allCharacteristic[i]-
allCharacteristic[i+1]>=4){
                    pick++;
                }
                break;
            case 6:
                if(allCharacteristic[i]-
allCharacteristic[i-1]>=4){
                    pick++;
                }
                break;
            default:
                if((allCharacteristic[i]-
allCharacteristic[i+1]>=2)&&(allCharacteristic[i]-
allCharacteristic[i-1]>=2)){
                    pick++;
                }
                break;
        }
    }
    switch(pick){
        case 4:
            Rezult.text += " Импульсивный";
            return "C";
            break;
        case 3:
            Rezult.text += " Импульсивный";

```

```

        return "C";
        break;
    case 2:
        Result.text += " Экспрессивный";
        return "D";
        break;
    case 1:
        float sum=0;
        for(int i =0;
i<allCharacteristic.Length; i++){
            sum+=allCharacteristic[i];
        }
        if(sum>=42){
            Result.text += " Уплощенный,
мотивационно всеяден";
            return "F";
        }else{
            Result.text += " Уплощенный,
мотивационно равнодушен";
            return "G";
        }
        break;
    case 0:
        float sum1=0;
        for(int i =0;
i<allCharacteristic.Length; i++){
            sum1+=allCharacteristic[i];
        }
        if(sum1>=42){

```



```

        Result.text += " Уплощенный,
мотивационно всеяден";
        return "F";
    }else{
        Result.text += " Уплощенный,
мотивационно равнодушен";
        return "G";
    }
    break;
}
}
return "";
}
}

```

После вывода диаграммы на экран функция “Start()” продолжает работать и выводит на экран данные о типе мотивационных профилей в различных сферах жизнедеятельности человека и уникальный код, который будет передан в базу данных пользователей прошедших тест. (на рисунке 10 это информация на белом фоне).

Уникальный код, представляющий из себя числовой вектор с присоединенными значениями имени, факультета, курса, пола и группы испытуемого, генерируется при помощи следующих строк в функции “Start()”.

```

float[] allCharacteristic = GetAllCharacteristic(4, 0);
string code =
CheckProgressivRegressiv(allCharacteristic, ResultGeneral);

```

```

code +=
CheckImpulsiveExtrovertedFlattened(allCharacteristic,
RezultGeneral);
    allCharacteristic = GetAllCharacteristic(2, 0);
code += CheckProgressivRegressiv(allCharacteristic,
RezultLifeStyle);
code +=
CheckImpulsiveExtrovertedFlattened(allCharacteristic,
RezultLifeStyle);
    allCharacteristic = GetAllCharacteristic(2, 2);
code += CheckProgressivRegressiv(allCharacteristic,
RezultStudy);
code +=
CheckImpulsiveExtrovertedFlattened(allCharacteristic,
RezultStudy);
    allCharacteristic = GetAllCharacteristic(1, 0);
code += CheckProgressivRegressiv(allCharacteristic,
TestText);
code +=
CheckImpulsiveExtrovertedFlattened(allCharacteristic,
TestText);
    allCharacteristic = GetAllCharacteristic(1, 1);
code += CheckProgressivRegressiv(allCharacteristic,
TestText);
code +=
CheckImpulsiveExtrovertedFlattened(allCharacteristic,
TestText);
    allCharacteristic = GetAllCharacteristic(1, 2);

```

```

        code += CheckProgressivRegressiv(allCharacteristic,
TestText);
        code +=
CheckImpulsiveExtrovertedFlattened(allCharacteristic,
TestText);
        allCharacteristic = GetallCharacteristic(1, 3);
        code += CheckProgressivRegressiv(allCharacteristic,
TestText);
        code +=
CheckImpulsiveExtrovertedFlattened(allCharacteristic,
TestText);
        code += CheckEmojiStatus(EmojiStatus);
        for(int i =0; i<32;i++){
            if(PlayerPrefs.GetInt(characteristic[i]) > 9){
                code+=" " +
PlayerPrefs.GetInt(characteristic[i]);
            }else{
                code+="0" +
PlayerPrefs.GetInt(characteristic[i]);
            }
        }
        code+="_"+PlayerPrefs.GetString("FullName")
+"_"+PlayerPrefs.GetString("Gender")+ "_"+
PlayerPrefs.GetString("Faculty") + "_" +
PlayerPrefs.GetString("Class")+ "_" +
PlayerPrefs.GetString("SubClass");
        ItogCode.text = code;

```

Который можно скопировать при помощи кнопки в левом нижнем углу.

Также на рисунке 9 есть еще две кнопки. Первая с надписью “Информация о мотивационных профилях”, откроет сводку с информацией о всех характеристиках, мотивационных и эмоциональных профилях — рисунок 10.



Рисунок 10 — Экраны с дополнительной информацией

Вторая кнопка с надписью “Сбросить все и пройти тест заново” сбрасывает все переменные и загружает самую первую сцену с рисунка 9.

### 3.2 Алгоритм работы сайта

Для обработки данных после прохождения теста создан сайт. Файл, состоящий из уникальных кодов, сортируется по выбранному параметру и выводит таблицу из этих пользователей — рисунок 11.

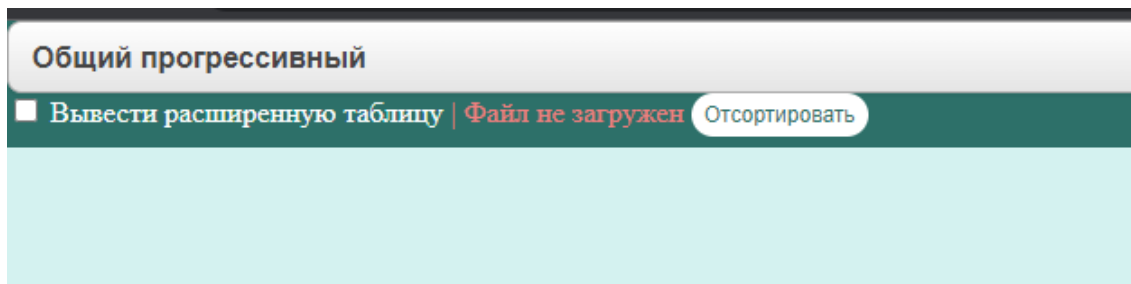


Рисунок 11 — Страница с обработкой результатов

Для того чтобы загрузить файл предусмотрена кнопка. На рисунке 12 это кнопка в виде знака плюс, расположенная снизу сайта. При нажатии откроется окно для выбора файла с компьютера пользователя. После полной загрузки базы данных сайт меняет статус с “Файл не загружен” на “Файл загружен и готов к обработке”, а также даст доступ к кнопке “Отсортировать”.

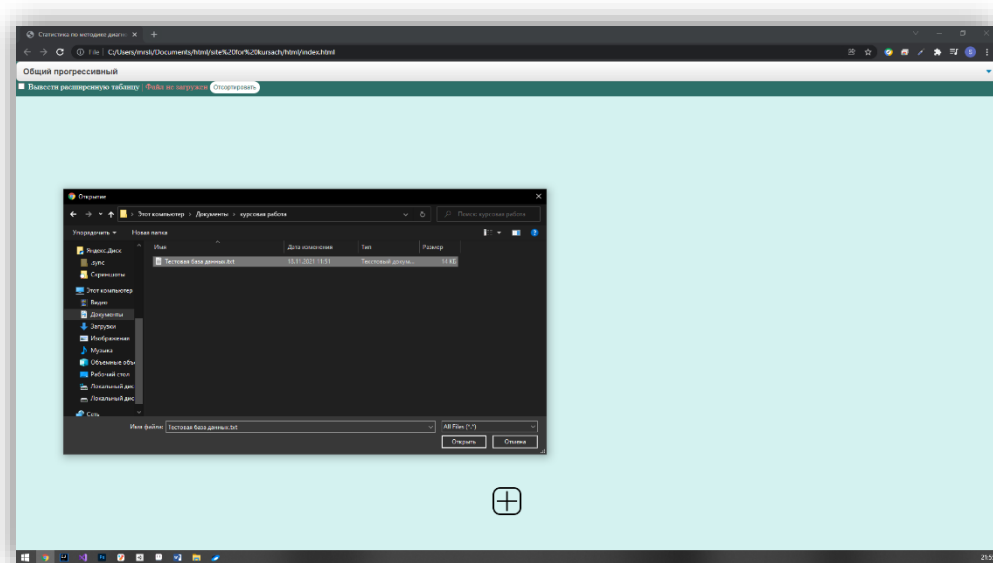


Рисунок 12 — Страница с открытым меню с выбором файла

Данная технология работает из-за html тега:

```
<input type="file" onchange="download(this)" />
```

Замаскированного под кнопку при помощи подключенного файла css. Где html — язык разметки, определяющий структуру и содержимое страницы [8], CSS — язык для определения правил стилизации элементов документа [9]. При нажатии на кнопку запускается javascript файл с функцией “download()”. Где javascript — скриптовый язык для определения динамического обновления содержимого страницы [10].

После чего из списка пользователь выбирает нужный ему вариант сортировки. После чего может отметить настройку, из-за которой будет выведена расширенная таблица со всеми данными — рисунок 13.

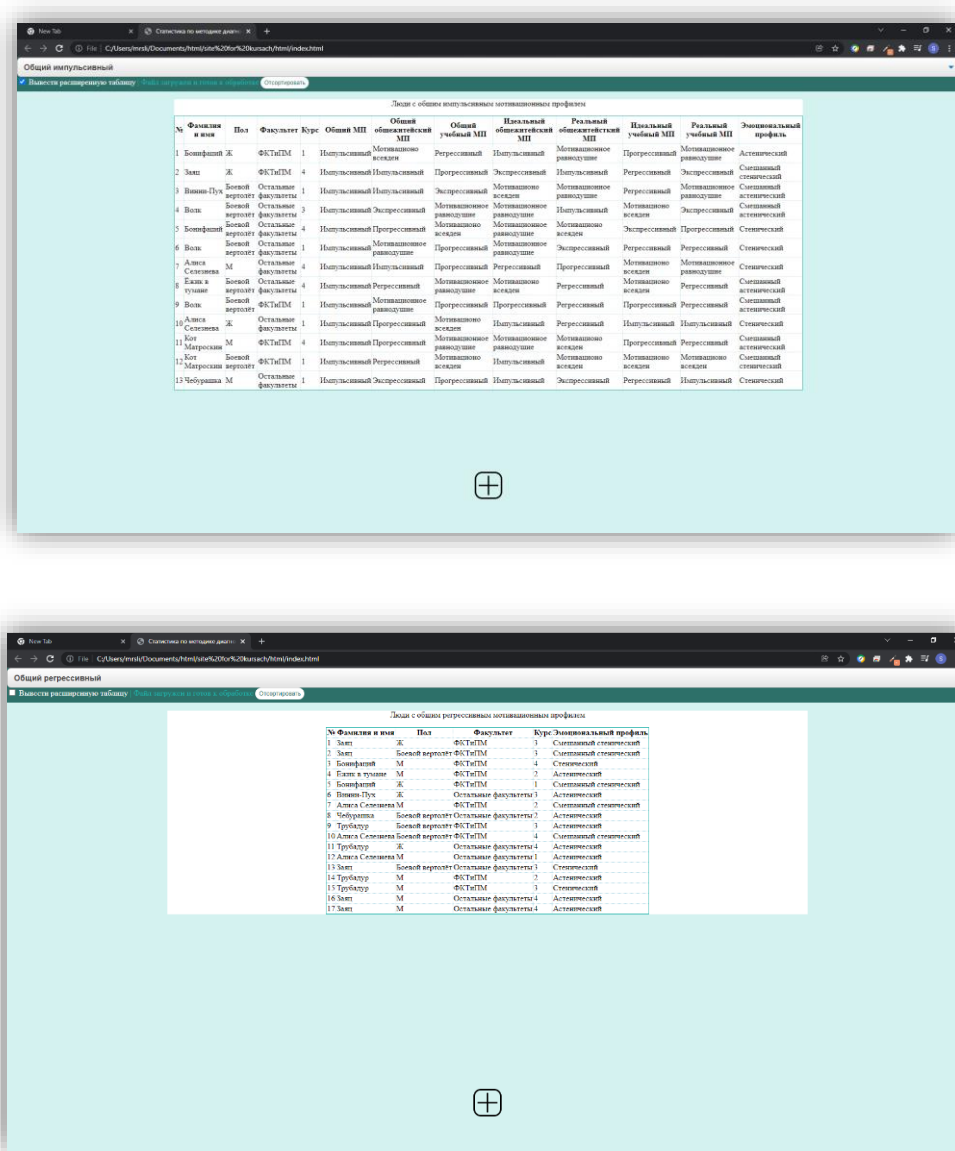


Рисунок 13 — Страницы с примерами вывода таблиц

## ЗАКЛЮЧЕНИЕ

Цель курсовой работы — разработка мобильного модуля психологического тестирования мотивационного профиля личности и web-страница для просмотра результатов, достигнута.

В курсовой работе рассмотрены основные подходы к понятию «мотивационный профиль личности». Обосновывается необходимость проведения диагностики мотивационного профиля сотрудников в современной организации. Приводятся актуальные методики диагностики мотивационного профиля личности таких авторов, как В.Э. Мильман, В.И. Герчиков, Э. Шейн, Ш. Ричи и П. Мартин, А.А. Меграбаян, Дж. Барбуто и Р. Сколла. Подробно изучена методика «диагностика мотивационной структуры личности» В.Э. Мильмана: суть, инструментарий, принцип применения в мобильной разработке. Проведен реферативный обзор существующих приложений, психологических тестов для мобильных устройств.

В работе предложен проект и реализация мобильного приложения, позволяющего провести психологический тест «Диагностики мотивационной структуры личности» В. Э. Мильмана, а также создан сайт для удобной работы с полученными данными. После прохождения теста результат трансформируется в уникальный код, который загружается в файл для дальнейшего анализа. Для построения запросов к данным разработана простая web-страница. Приложение для психологического теста разработано на языке C# с использованием Unity. Страница разработана с использованием технологий html, css и javascript. Программа нужна для выявления мотивационных профилей сотрудников и реорганизации на основании полученных данных для улучшения продуктивности предприятия.

Развитие данной темы заключается в тестировании методики на реальных людях, и выведении прогноза точности этих мотивационных профилей. Возможны не существенные улучшения дизайна и изменение интерфейса программы в более дружелюбном ключе для конечного

пользователя. Создание полноценной базы данных в которую будут автоматически присылаться данные пользователя после прохождения им теста.



## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Ричи, Ш. Управление мотивацией. / Ш. Ричи, П. Мартин — Москва: ЮНИТИ-ДАНА, 2004. — 399 с. — ISBN 5-238-00599-7
2. Мильман, В.Э. Метод изучения мотивационной сферы личности: Практикум по психодиагностике. Психодиагностика мотивации и саморегуляции / В. Э. Мильман. — Москва : МГУ М, 1990. — С.23-43.
3. Яхудина, Е.Н. Мотивационная структура и удовлетворенность трудом воспитателей ДОО / Е.Н. Яхудина, Ю.В. Вдовина — Санкт-Петербург : Царскосельские чтения, 2017. — С. 373-376.
4. Википедия // IOS: [Электронный ресурс] — 2021. URL: <https://ru.wikipedia.org/wiki/IOS> (дата обращения: 30.11.2021).
5. Википедия // Android: [Электронный ресурс] — 2021. URL: <https://ru.wikipedia.org/wiki/Android> (дата обращения: 30.11.2021).
6. Хокинг, Д. Unity в действии : Мультиплатформенная разработка на C# /Д. Хокинг — Санкт-Петербург : Издательский Дом ПИТЕР, 2018. — 352 с. — ISBN: 978-5-4461-0816-9
7. Либерти, Д. Язык программирования C# / Программирование на C#. //Санкт-Петербург. Символ-Плюс — 2003. — ISBN 5-93286-038-3. — 688 с.
8. Гоше, Х. Д. HTML5 : Для профессионалов / Х. Д. Гоше — Санкт-Петербург : Издательский Дом ПИТЕР, 2014. — 496 с. — ISBN: 978-5-496-01397-0
9. Сойер, М. Д. Новая большая книга CSS / М. Д. Сойер — Санкт-Петербург : Издательский Дом ПИТЕР, 2018 — 720 с. — ISBN: 978-5-496-02080-0
10. Флэнаган, Д. JavaScript Полное руководство / Д. Флэнган — Справочник по самому популярному языку программирования — Москва : Символ-Плюс, 2019 — ISBN: 978-5-93286-215-5 — 720 с.