



eISSN 2311-2468
Том 10, № 14. 2022
Vol. 10, no. 14. 2022

электронное периодическое издание
для студентов и аспирантов

Огарёв-онлайн Ogarev-online

<https://journal.mrsu.ru>



ГУЩИНА О. А., ГУЩИН А. В.
РЕШЕНИЕ ЗАДАЧИ СОЗДАНИЯ АНСАМБЛЯ
СЛУЧАЙНОГО ЛЕСА ДЛЯ КЛАССИФИЦИРОВАНИЯ
МОДЕЛЕЙ ТЕЛЕФОНОВ ПО ЦЕНОВЫМ КАТЕГОРИЯМ

Аннотация. В статье рассматривается создание и использование ансамблей случайного леса деревьев решений для предсказательной аналитики на примере решения задачи классификации восьмидесяти моделей телефонов с двадцатью признаками по четырем ценовым категориям. Проведен сравнительный анализ полученного решения с решением на основе метода деревьев решений.

Ключевые слова: дерево решений, случайный лес решений, предсказательная аналитика.

GUSHCHINA O. A., GUSHCHIN A. V.
SOLUTION TO THE PROBLEM OF CREATING A RANDOM FOREST ENSEMBLE
FOR CLASSIFICATION OF PHONE MODELS BY PRICE CATEGORIES

Abstract. The article discusses the creation and use of random forest ensembles of decision trees for predictive analytics by the example of solving the problem of classifying eighty phone models with twenty features in four price categories. A comparative analysis of the obtained solution with a solution based on the decision tree method was also carried out.

Keywords: decision tree, random decision forest, predictive analytics.

Постановка задачи. Пусть даны 80 различных моделей телефонов различных марок. Каждой модели телефона поставлено в соответствие уникальное сочетание из 20 признаков: емкость телефонной батарейки; наличие Bluetooth; скорость процессора; наличие разъёма для второй SIM-карты; число мегапикселей в фронтальной камере; поддержка 4G; размер внутренней памяти; толщина телефона; вес телефона; количество ядер процессора; количество мегапикселей в основной камере; объем оперативной памяти; высота экрана; ширина экрана; высота телефона; ширина телефона; время разговора от одной батареи; поддержка 3G; наличие сенсорного экрана; поддержка wifi.

На основе этих данных необходимо для каждой модели телефона спрогнозировать ценовую категорию (низкая, средняя, высокая и очень высокая).

Процесс решения задачи. Для хранения данных используется dataframe из библиотеки Pandas [1; 2] и массив из библиотеки NumPy, которые подключаются в начале файла. После загрузки данных можно просмотреть их, и их статистическую информацию. Результаты этих манипуляций представлены в таблицах 1–4.

Таблица 1

Часть элементов выборки

№	battery_power	blue	clock_speed	...	touch_screen	wifi	price_range
0	842	0	2.2	...	0	1	1
1	1021	1	0.5	...	1	0	2
2	563	1	0.5	...	1	0	2
3	615	1	2.5	...	0	0	2
4	1821	1	1.2	...	1	0	1
...							
1995	794	1	0.5	...	1	0	0
1996	1965	1	2.6	...	1	1	2
1997	1911	0	0.9	...	1	0	3
1998	1512	0	0.9	...	1	1	0
1999	510	1	2.0	...	1	1	3

Таблица 2

Типы всех использованных переменных

battery_power, Blue, dual_sim, Fc, four_g, int_memory, mobile_wt, n_cores, Pc, px_height, px_width, Ram, sc_h, sc_w, talk_time, three_g, touch_screen, touch_screen, wifi, price_range	int64
clock_speed, m_dep	float64

Таблица 3

Результаты описательной статистики

№	battery_power	blue	...	wifi	price_range
count	2000.000000	2000.0000	...	2000.000000	2000.000000
mean	1238.518500	0.4950	...	0.507000	1.500000
std	439.418206	0.5001	...	0.500076	1.118314
min	501.000000	0.0000	...	0.000000	0.000000
25%	851.750000	0.0000	...	0.000000	0.750000
50%	1226.000000	0.0000	...	1.000000	1.500000
75%	1615.250000	1.0000	...	1.000000	2.250000
max	1998.000000	1.0000	...	1.000000	3.000000

Категориальные и количественные переменные

Категориальные	[]
Количественные	['battery_power', 'blue', 'clock_speed', 'dual_sim', 'fc', 'four_g', 'int_memory', 'm_dep', 'mobile_wt', 'n_cores', 'pc', 'px_height', 'px_width', 'ram', 'sc_h', 'sc_w', 'talk_time', 'three_g', 'touch_screen', 'wifi', 'price_range']

Перед передачей данных в будущую модель их разделили на признаки и классы. После разделения провели распределение данных на обучающую и тестовую выборки с параметром `test_size=0.25`, то есть три четверти данных передали в обучающую выборку, а одну четверть – в тестовую.

На следующем этапе воспользовались библиотекой `scikit-learn` для создания объекта классификатора случайного леса и выполнения его обучение на представленных данных [3-6]. Переменную `params` используем для хранения параметров ансамбля. Эти данные передаем в функцию создания экземпляра случайного леса решений, фрагмент которого представлен далее. Далее воспользовались функцией обучения с передачей тренировочных данных.

Для контроля результата после обучения модели случайного леса оценили следующие показатели эффективности полученного решения на обучающей и тестовой выборках: точность на обучающей выборке, точность контрольной выборки, OOB-оценка, AUC на обучающей выборке и AUC на контрольной выборке. Результат приведен в таблице 5.

Таблица 5

Результаты вычисления точности модели случайного леса решений

Метрика	Результат
Правильность на обучающей выборке	0.999
Правильность на контрольной выборке	0.880
OOB оценка правильности	0.89
AUC на обучающей выборке для RandomForest	1.000
AUC на контрольной выборке для RandomForest	0.982

В качестве примера спрогнозировали принадлежность первых пяти объектов к определенному классу с помощью функции `forestClassifier.predict_proba(X_test[:5]))`. Результат представлен в таблице 6. Каждый столбец соответствует определенному классу (ценовой категории), а строка – определенной модели телефона.

Пример вывода спрогнозированных вероятностей принадлежности первых пяти модели телефона (таблица 1) к одной из четырех ценовых категорий

0.01039387	0.06722729	0.15937753	0.76300131
0.0517534	0.09618108	0.26739106	0.58467446
0.06692121	0.44428379	0.37681018	0.11198482
0.01541764	0.04722354	0.13160823	0.80575059
0.23304687	0.55828515	0.15961096	0.04905703

Дополнительно с вычислением метрики точности модели случайного леса решений, была вычислена матрица ошибок (рис. 1) и сформирован квалификационный отчет (рис. 2).

Матрица ошибок:
[[112 6 0 0]
[13 102 11 0]
[0 20 105 9]
[0 0 11 111]]

Рис. 1. Матрица ошибок.

Классификационный отчет:

	precision	recall	f1-score	support
0.0	0.90	0.95	0.92	118
1.0	0.80	0.81	0.80	126
2.0	0.83	0.78	0.80	134
3.0	0.93	0.91	0.92	122
accuracy			0.86	500
macro avg	0.86	0.86	0.86	500
weighted avg	0.86	0.86	0.86	500

Рис. 2. Квалификационный отчет.

Преимуществом случайного леса и дерева решений перед другими методами прогнозирования является возможность оценивания влияния определенных признаков на конечный результат. Для этой цели был введен параметр классификатора случайного леса решений в виде одномерного массива. Для увеличения его информативности пришлось отсортировать все признаки по степени влияния на модель. Результат приведен в таблице 7.

Таблица 7

Показатели важности признаков случайного леса решений

1	ram	0.522775
2	battery_power	0.076009
3	px_height	0.055573
4	px_width	0.054512
5	mobile_wt	0.035392
6	int_memory	0.031938
7	talk_time	0.026818
8	pc	0.026318
9	clock_speed	0.025707
10	sc_w	0.025151
11	sc_h	0.024550
12	fc	0.021197
13	m_dep	0.020963
14	n_cores	0.019568
15	dual_sim	0.005946
16	wifi	0.005899
17	touch_screen	0.005887
18	four_g	0.005619
19	blue	0.005320
20	three_g	0.004857

Чтобы понять, насколько эффективен ансамблевый подход мы построили дерево решений, обучили на том же наборе данных и вычислили для него показатели эффективности. Результат оценки модели дерева решений различными метриками представлен на таблице 8.

Таблица 8

Результаты вычисления точности модели дерева решений

Метрика	Результат
Правильность на обучающей выборке	1.000
Правильность на контрольной выборке	0.836
AUC на обучающей выборке для RandomForest	1.000
AUC на контрольной выборке для RandomForest	0.876

Сравнив показатели эффективности обучения для случайного леса и дерева решений в таблицы 5 и 8, отметим, что точность первого больше на 0.044, а значение AUC на контрольной выборке лучше на 0.106. Следовательно, ансамбль деревьев решений эффективнее дерева решений.

Для анализа показателей важности признаков дерева решений оценивания влияния определенных признаков на конечный результат и получили данные таблицы 9. Из нее видно, что два последних признака (touch_screen и four_g) не влияют на прогноз (и для эффективности работы алгоритма их можно убрать). Так же, как и для случайного леса решений, для дерева решений абсолютное влияние имеет первый признак (ram).

Таблица 9

Показатели важности признаков дерева решений

1	ram	0.659793
2	battery_power	0.093422
3	px_height	0.091308
4	px_width	0.085851
5	mobile_wt	0.012314
6	talk_time	0.010347
7	sc_w	0.007392
8	sc_h	0.005382
9	fc	0.005496
10	pc	0.004757
11	m_dep	0.004418
12	n_cores	0.004289
13	three_g	0.003922
14	dual_sim	0.003846
15	clock_speed	0.003296
16	int_memory	0.002545
17	blue	0.001433
18	wifi	0.000989
19	touch_screen	0.000000
20	four_g	0.000000

Для понимания структуры дерева решений в ансамбле и в отдельности, мы их визуализировали и вывели в PNG-файл (рис. 3, 4).



Рис. 3. Визуализация дерева решений из ансамбля.

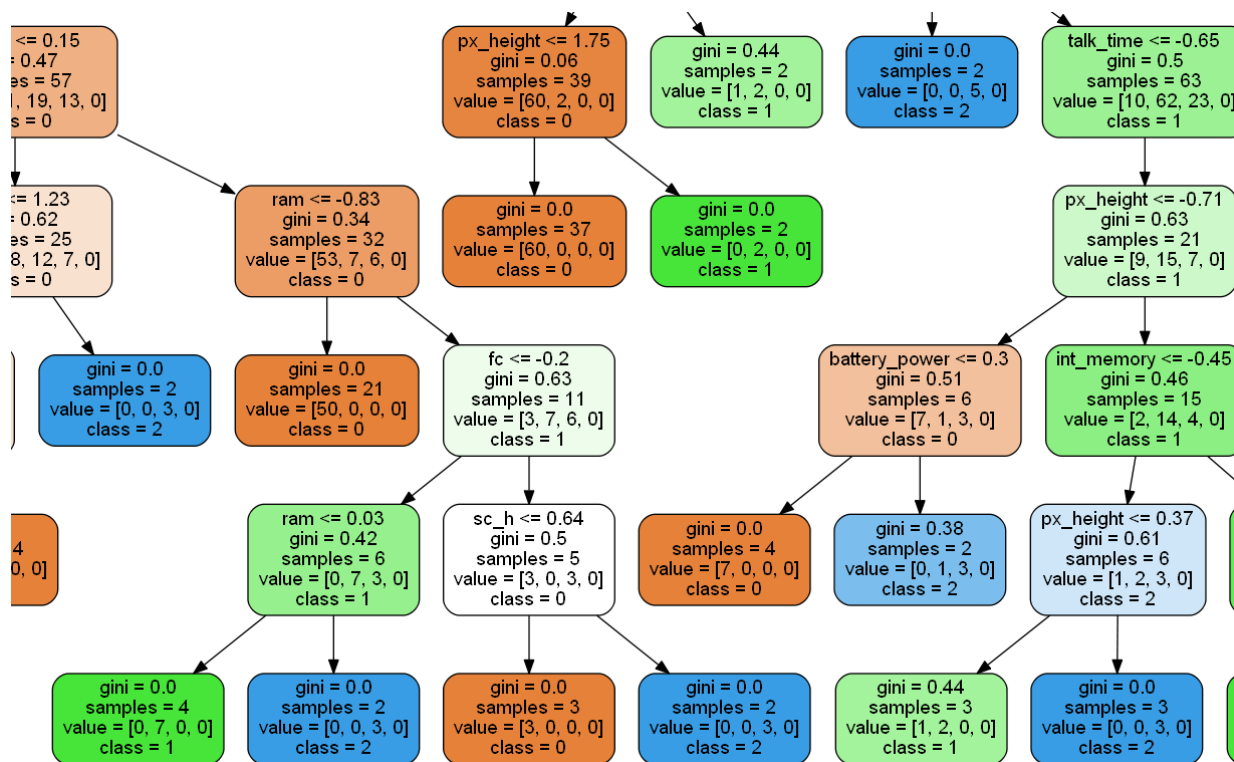


Рис. 4. Фрагмент дерева решений из ансамбля.

В каждом дереве решений узлы содержат: вопрос по одному признаку, по которому выполняется дальнейший выбор; неопределенность; количество прошедших образцов; отношение классов, прошедших через этот узел, выраженное в абсолютных числах; значение класса, большинство образцов которого прошли через узел (для листьев это прогнозируемое значение для всех элементов, попавших в него).

Так как листья являются заключительными прогнозируемыми значениями классификации, то они не содержат вопроса. При обработке нового элемента, нужно спускаться вниз по дереву, используя признаки элемента для ответа на вопросы. В конце элемент дойдет до одного из листьев и по значению листа определяется принадлежность к одному из классов.

Для повышения точности необходимо использовать оптимальные параметры. Самым простым способом их выявления считается метод сеточного поиска GridSearchCV с кросс-валидацией из библиотеки sklearn.model_selection. Суть метода состоит в том, что строит множество моделей с различными параметрами, обучающимися и сравнивающимися по эффективности. Оптимальным числом различных моделей ансамблей является произведение значения кросс-валидации и числа комбинаций выбранных параметров.

После прохода всех параметров, показанного на рисунке 5, получаем гиперпараметры для лучшей модели (рис. 6).

```
Fitting 3 folds for each of 4320 candidates, totalling 12960 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done  52 tasks      | elapsed:    3.8s
[Parallel(n_jobs=-1)]: Done 216 tasks      | elapsed:   19.3s
[Parallel(n_jobs=-1)]: Done 466 tasks      | elapsed:   42.5s
[Parallel(n_jobs=-1)]: Done 816 tasks      | elapsed:   1.2min
[Parallel(n_jobs=-1)]: Done 1266 tasks     | elapsed:   1.9min
[Parallel(n_jobs=-1)]: Done 1816 tasks     | elapsed:   2.7min
[Parallel(n_jobs=-1)]: Done 2466 tasks     | elapsed:   3.7min
[Parallel(n_jobs=-1)]: Done 3216 tasks     | elapsed:   4.9min
[Parallel(n_jobs=-1)]: Done 4066 tasks     | elapsed:   6.4min
[Parallel(n_jobs=-1)]: Done 5016 tasks     | elapsed:   8.0min
[Parallel(n_jobs=-1)]: Done 6066 tasks     | elapsed:  10.1min
[Parallel(n_jobs=-1)]: Done 7216 tasks     | elapsed:  12.5min
[Parallel(n_jobs=-1)]: Done 8466 tasks     | elapsed:  15.1min
[Parallel(n_jobs=-1)]: Done 9816 tasks     | elapsed:  18.0min
[Parallel(n_jobs=-1)]: Done 11266 tasks    | elapsed:  21.2min
[Parallel(n_jobs=-1)]: Done 12816 tasks    | elapsed:  24.7min
[Parallel(n_jobs=-1)]: Done 12960 out of 12960 | elapsed: 25.0min finished
```

Рис. 5. Результат вычисления гиперпараметров.

```
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                       criterion='gini', max_depth=15, max_features='sqrt',
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=3, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=700,
                       n_jobs=None, oob_score=True, random_state=None,
                       verbose=0, warm_start=False)
```

Рис. 6. Результат вывода гиперпараметров лучшей модели.

Проведя эксперимент по увеличению точности показателей в зависимости от максимальной глубины дерева и количества признаков в узле получили данные, приведенные в таблице 10.

Таблица 10

**Результаты показателей точности модели случайного дерева решений
с различными гиперпараметрами**

param_max_depth	param_max_features	Точность
2	log2	0.718581
	sqrt	0.719463

3	log2	0.775933
	sqrt	0.775521
7	log2	0.836848
	sqrt	0.837754
11	log2	0.844720
	sqrt	0.844389
15	log2	0.844227
	sqrt	0.845271

На рисунке 7 приведен график зависимости вероятности совпадения полученных решений с реальными данными от количества деревьев в случайном лесе решений.

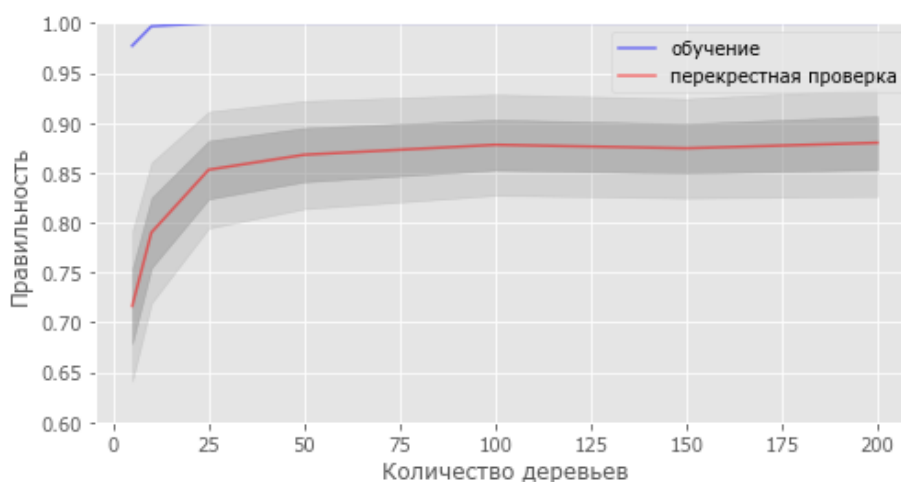


Рис. 7. График зависимости вероятности совпадения полученных решений с реальными данными от количества деревьев.

Заключение. В результате проведенного исследования были получены достаточно точные практические данные по предсказанию ценовой категории данных марок телефонов, а также сделан вывод о высокой эффективности использования ансамблей случайного леса и деревьев решений для предсказания классификации данных.

СПИСОК ЛИТЕРАТУРЫ

1. Ансамблевые методы: бэггинг, бустинг и стекинг [Электронный ресурс]. – Режим доступа: <https://neurohive.io/ru/osnovy-data-science/ansamblevye-metody-begging-busting-i-steking/> (дата обращения: 03.11.2022).
2. Груздев А. В. Прогнозное моделирование в IBM SPSS Statistics, R и Python: метод деревьев решений и случайный лес. – М.: ДМК Пресс, 2018. – 642 с.

3. Открытый курс машинного обучения. Тема 5. Композиции: бэггинг, случайный лес [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/company/ods/blog/324402> (дата обращения: 03.11.2022).
4. Джоши П. Искусственный интеллект с примерами на Python: Пер. с англ. – СПб.: ООО «Диалектика», 2019. – 448 с.
5. A random forest classifier [Электронный ресурс]. – Режим доступа: <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier> (дата обращения: 03.11.2022).
6. Hastie T. The Elements of Statistical Learning. – Springer, 2017. – 764 p.

КУТЫРКИНА М. А., МАМЕДОВА Т. Ф.
ЧИСЛЕННОЕ МОДЕЛИРОВАНИЕ НЕЙРОННОЙ СЕТИ
НА ОСНОВЕ МОДЕЛИ ХОДЖКИНА-ХАКСЛИ

Аннотация. Статья посвящена численному моделированию колебаний мембранного потенциала и воротных переменных в модели Ходжкина-Хаксли. Особое внимание уделяется исследованию изменения входного тока на нейронах, от которого зависит способность нейронов к обучению и запоминанию информации.

Ключевые слова: математическая модель Ходжкина-Хаксли, нейронная сеть, мембранная емкость, воротные переменные.

KUTYRKINA M. A., MAMEDOVA T. F.
NUMERICAL SIMULATION OF NEURAL NETWORK
BASED ON HODGKIN-HUXLEY MODEL

Abstract. The article deals with numerical simulation of membrane potential fluctuations and gate variables in the Hodgkin-Huxley model. Particular attention is paid to the study of changes in the input current on neurons, which determines the ability of neurons to learn and remember information.

Keywords: Hodgkin-Huxley mathematical model, neural network, membrane capacitance, gate variables.

Актуальность проблемы. Современная неврология предъявляет высокие требования к квалификации врачей и методам лечения. Внедрение в неврологию математических методов, в частности методов математического моделирования, позволяет существенно расширить прогнозирование, диагностику и терапию заболеваний.

Нейрон — основной функциональный элемент нервной системы. Типичный нейрон состоит из трех частей: тела, дендритов и аксона. Дендрит принимает сигналы от другого нейрона, а аксон передает их другим нейронам. Место контакта аксона одного нейрона и дендрита другого нейрона называется синапсом. Сигналы, принимаемые с дендритов, суммируются в теле нейрона и если данная сумма превышает определённый порог, то возникает нервный импульс или спайк [1]. Кроме возбуждения нейроны могут и тормозить друг друга, уменьшая "сумму" внутри нейрона и препятствуя возникновению спайка.

Оболочка нейрона является хорошим изолятором, поэтому ионные составы вне и внутри нейрона различны. Заряженные частицы переносятся между внутренностью нейрона и окружающей его межклеточной жидкостью с помощью так называемых ионных насосов (существует, например, натрий-калиевый насос). Импульсы от возбуждающих и тормозящих

нейронов, приходящие к нашему нейрону, "включают" разные виды ионных насосов. Так регулируется электрическое поле нейрона. В зависимости от того, каким оно будет, импульс либо победит дальше, либо нет.

Постановка задачи. На данный момент созданы различные как относительно простые модели, например, «Integrate and Fire», в которой нейрон представляется в виде конденсатора и резистора, так и более детальные, биологически правдоподобные, модели, например, модель Ходжкина-Хаксли, которая гораздо сложнее как в вычислительном плане, так и в плане анализа её динамики, но она гораздо точнее описывает динамику мембранного потенциала нейрона [2].

Классическая модель Ходжкина-Хаксли описывается следующей системой дифференциальных уравнений [3]:

$$\begin{cases} C_m \frac{dV}{dt} = I_{ext} + I_{syn}(t) - G_{Na} - G_K - G_L, \\ \frac{dm}{dt} = \alpha(V)(1 - m) - \beta_m(V)m, \\ \frac{dh}{dt} = \alpha_h(V)(1 - h) - \beta_h(V)h, \\ \frac{dn}{dt} = \alpha_n(V)(1 - n) - \beta_n(V)n, \end{cases}$$

где

$$\begin{aligned} \alpha_m(V) &= \frac{0.1(25 - V)}{\frac{\exp(25 - V)}{10} - 1}; & \alpha_n(V) &= \frac{0.01(10 - V)}{1 + \exp\left(\frac{10 - V}{10}\right)}; \\ \beta_m(V) &= 4 \exp\left(-\frac{V}{18}\right); & \beta_n(V) &= 0.125 \exp\left(-\frac{V}{80}\right); \\ \alpha_h(V) &= 0.07 \exp\left(-\frac{V}{20}\right); & G_{Na} &= g_{Na}m^3h(V - V_{Na}); \\ \beta_h(V) &= \frac{1}{1 + \exp\left(\frac{30 - V}{10}\right)}; & G_K &= g_Kn^4(V - V_K); \\ & & G_L &= g_L(V - V_L). \end{aligned}$$

Начальные условия для системы (1) имеют вид

$$V(0) = V_0; m(0) = m_0; h(0) = h_0; n(0) = n_0.$$

Здесь $V(t)$ – мембранный потенциал, $m(t)$, $h(t)$, $n(t)$ – так называемые воротные переменные: m , n – активационные переменные для калиевых и натриевых каналов, h – инактивационная переменная натриевых каналов. Эти переменные задают изменения проводимости ионных каналов в зависимости от мембранного потенциала, C_m – мембранная емкость. Остальные параметры были получены экспериментальным путем и имеют следующие значения:

$$g_{Na} = 120 \text{ мСм}; g_K = 36 \text{ мСм}; g_L = 0,3 \text{ мСм};$$

$$V_{Na} = 115 \text{ мВ}; V_K = -12 \text{ мВ}; V_L = 10,6 \text{ мВ}.$$

Сумма синаптических токов рассчитывается по следующей формуле:

$$I_{syn} = C_m g_{exc} \left[\sum_{k=1}^{N_e} \sum_I \mathbf{1}(t - t_k^I) - K \sum_{m=1}^{N_i} \sum_n \mathbf{1}(t - t_m^n) \right].$$

Здесь N_e – количество возбуждающих нейронов, N_i – количество тормозных нейронов, $\mathbf{1}(t - t_k^I)$ и $\mathbf{1}(t - t_m^n)$ – индикаторные функции, принимающие значение 1, если их аргументы равны нулю, и 0 – в противном случае, t_k^I – время разряда I -го спайка k -го нейрона, C_m – мембранная емкость. Константы, $g_{exc} = 1$ и $K = 4$ вычислены эмпирически.

Численные расчеты получены методом Рунге-Кутты 4-го порядка при $t \in [0, 300]$ и следующих параметрах: количество возбуждающих нейронов – 200, количество тормозящих нейронов – 50.

Данная модель была реализована в среде разработки Google Colaboratory. При программировании использовались библиотеки NumPy и SciPy, а также библиотека Matplotlib, позволяющая визуализировать полученные данные.

Рассмотрим случай, когда значение входного тока I_{ext} принимает значение, равное 5 мА.

На рис. 1 представлен график колебаний мембранного потенциала с течением времени, на рис. 2 – график изменения воротных переменных с течением времени, и на рис. 3 – зависимость мембранного потенциала V от времени.

Исходя из данных графиков, можно сделать вывод, что при значении входного тока $I_{ext} = 5 \text{ мА}$ колебания мембранного потенциала исчезают. То же происходит и с воротными переменными. Из рис. 3 видно, что траектория зависимости воротных переменных от мембранного потенциала V не является периодической.

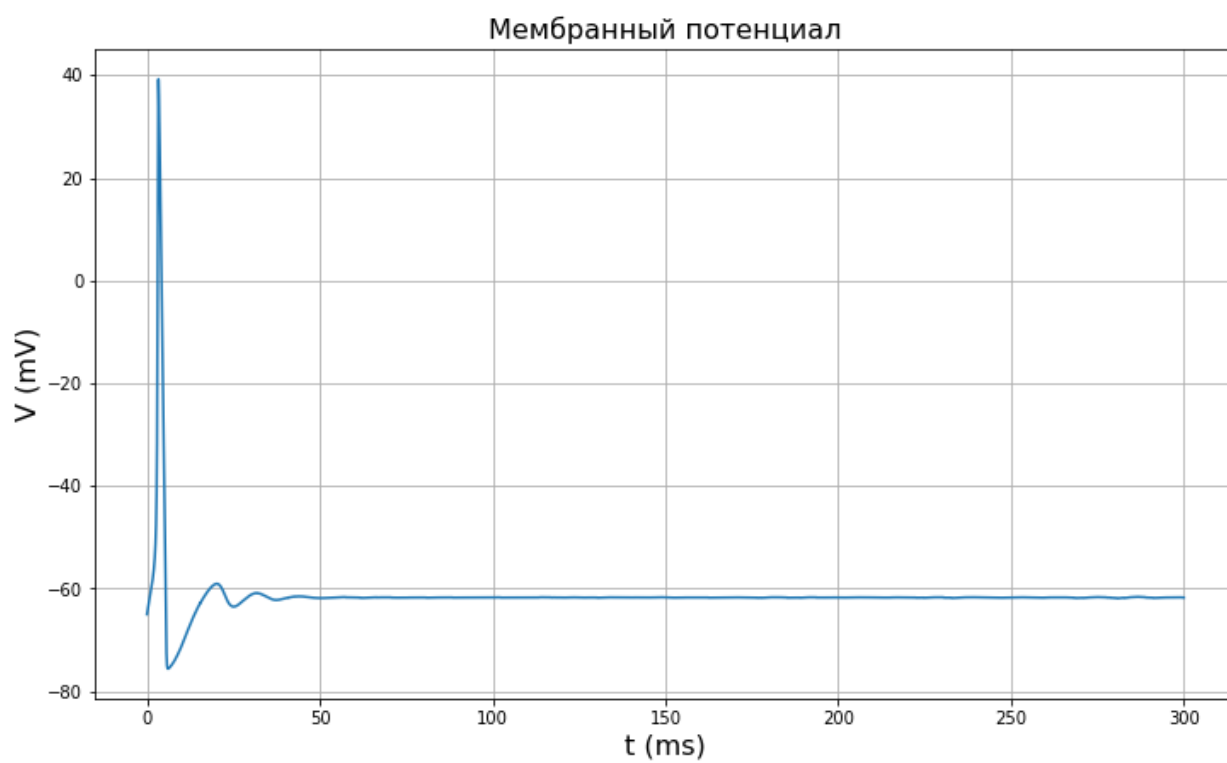


Рис. 1. Зависимость мембранного потенциала V от времени по модели Ходжкина-Хаксли при значении входного тока $I_{ext} = 5$ мА.

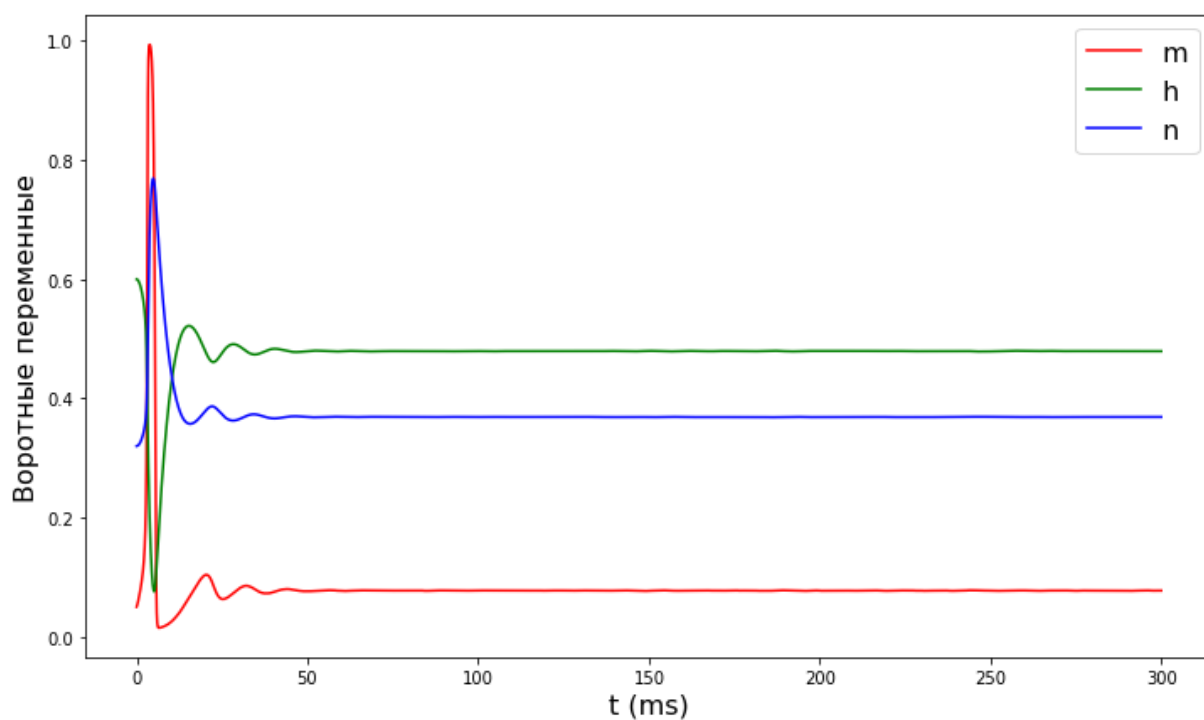


Рис. 2. Изменение воротных переменных с течением времени при значении входного тока $I_{ext} = 5$ мА.

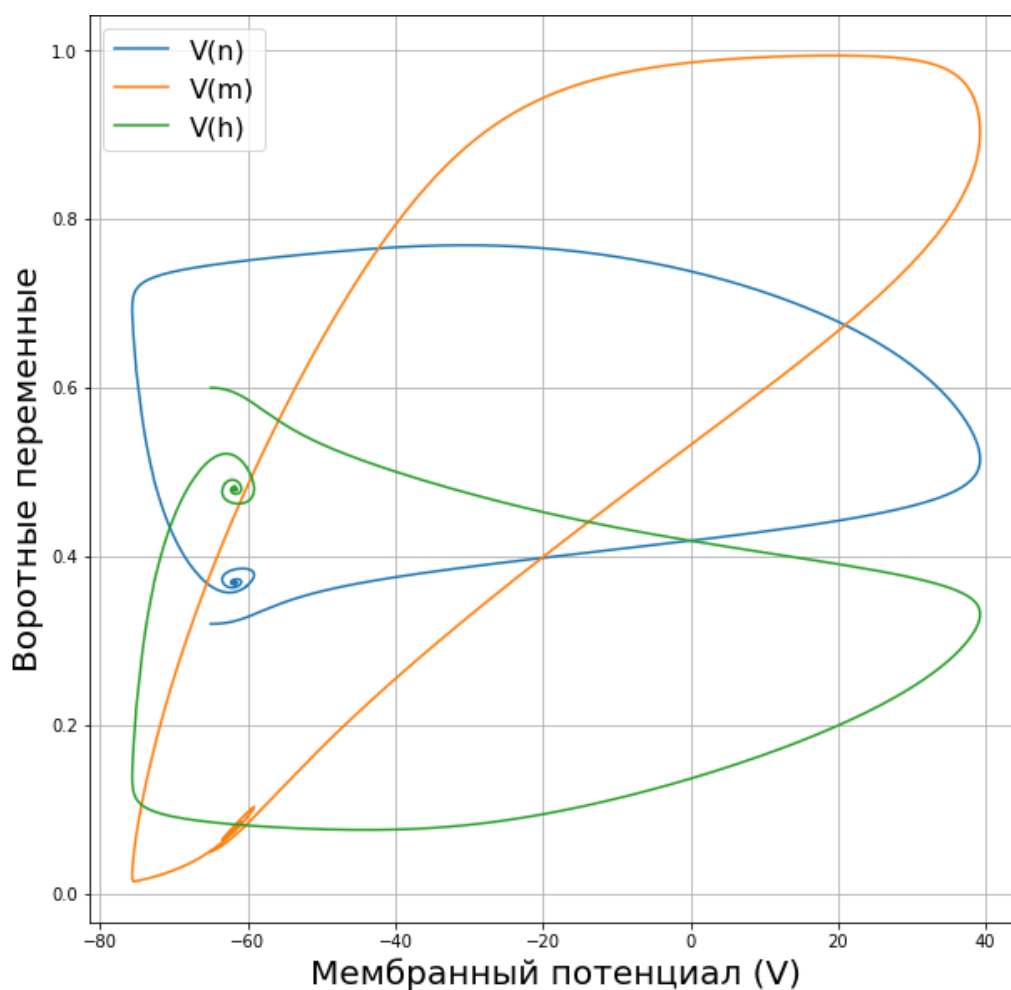


Рис. 3. Зависимость воротных переменных от мембранного потенциала V при значении входного тока $I_{ext} = 5$ мА.

С другой стороны, если входной ток достигает значения 5 мА, то предельный цикл фактически вырождается. Это значит, что такое значение входного тока слишком мало, чтобы возбудить нейрон.

Увеличим значение входного тока до 10 мА. По графику колебаний мембранного потенциала, представленного на рис. 4, и графику воротных изменения воротных переменных (см. рис. 5), можно предположить, что процесс является периодическим. После 1-2 пиков потенциал стабилизируется. Аналогичным образом ведут себя воротные переменные (см. рис. 5). Можно предположить (см. рис. 6), что траектория зависимости воротных переменных от мембранного потенциала V представляет собой невырожденный предельный цикл.

Судя по результатам численного эксперимента, при значении входного тока в 10 мА имеем наиболее ярко выраженный предельный цикл. Данное значение является оптимальным для возбуждения нейрона и передачи сигналов от одного нейрона другим.

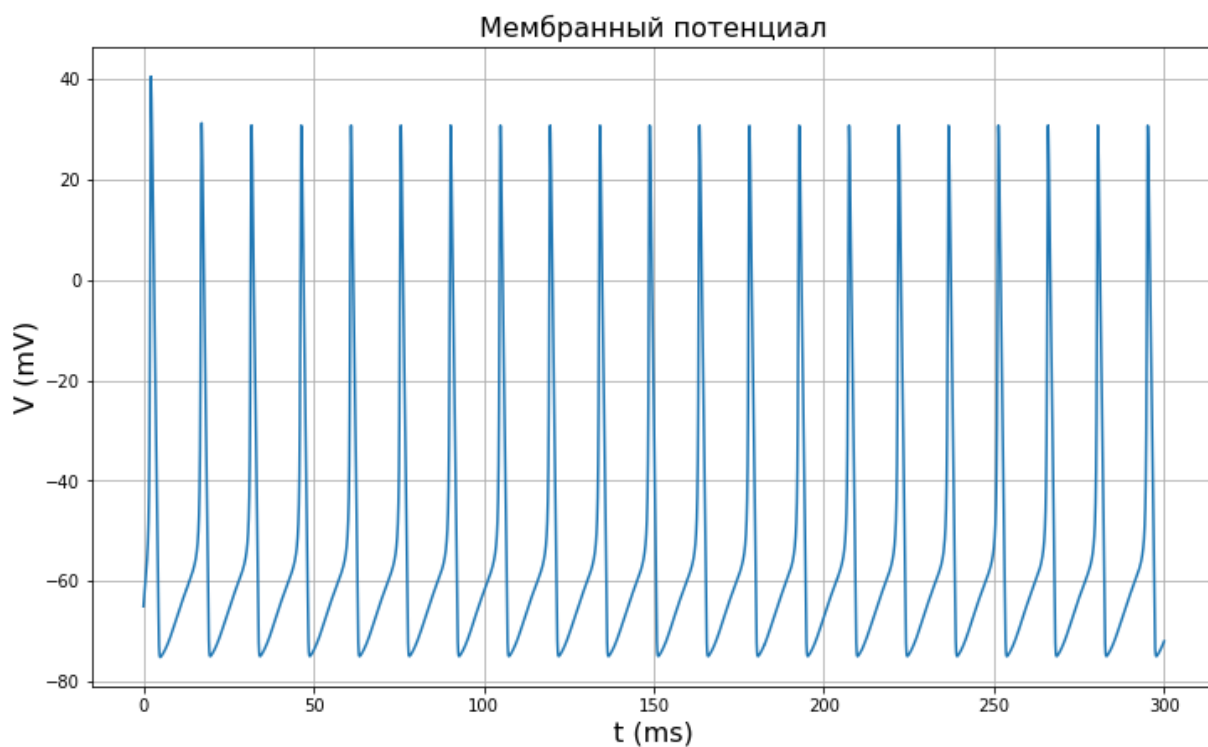


Рис. 4. Зависимость мембранного потенциала от времени по модели Ходжкина-Хаксли при значении входного тока $I_{ext} = 10$ мА.

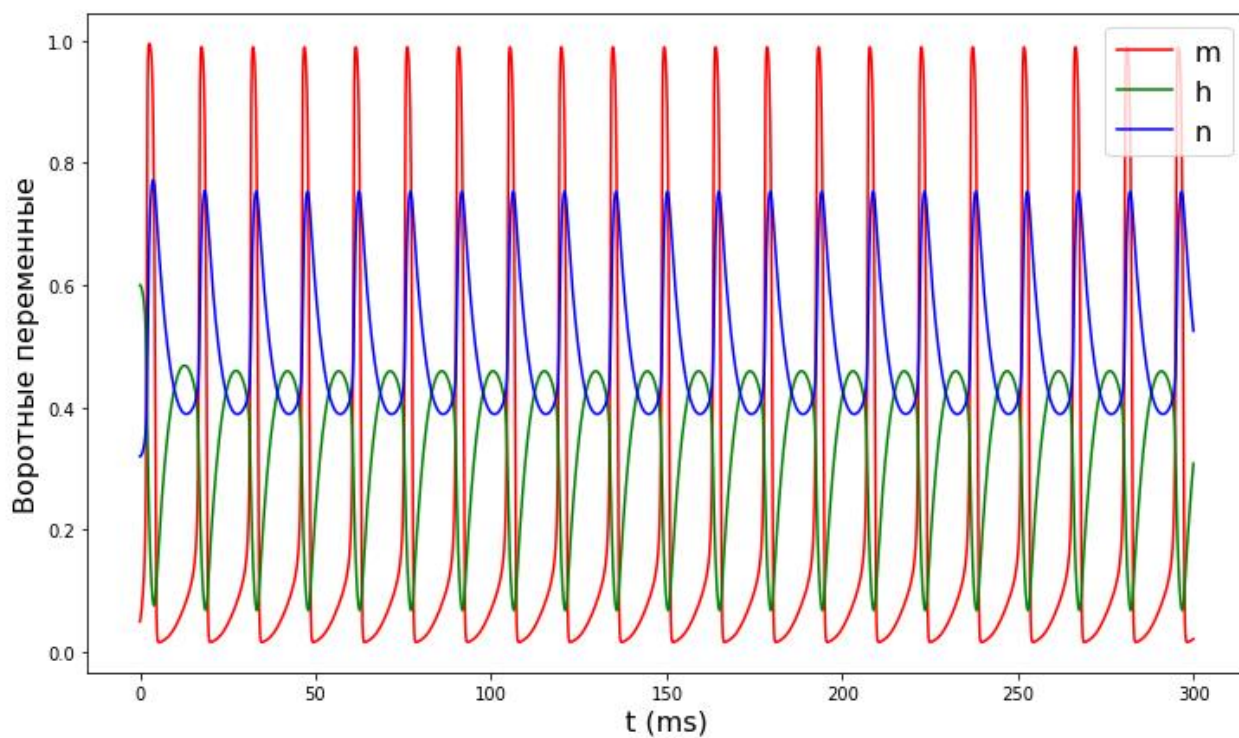


Рис. 5. Изменение воротных переменных с течением времени при значении входного тока $I_{ext} = 10$ мА.

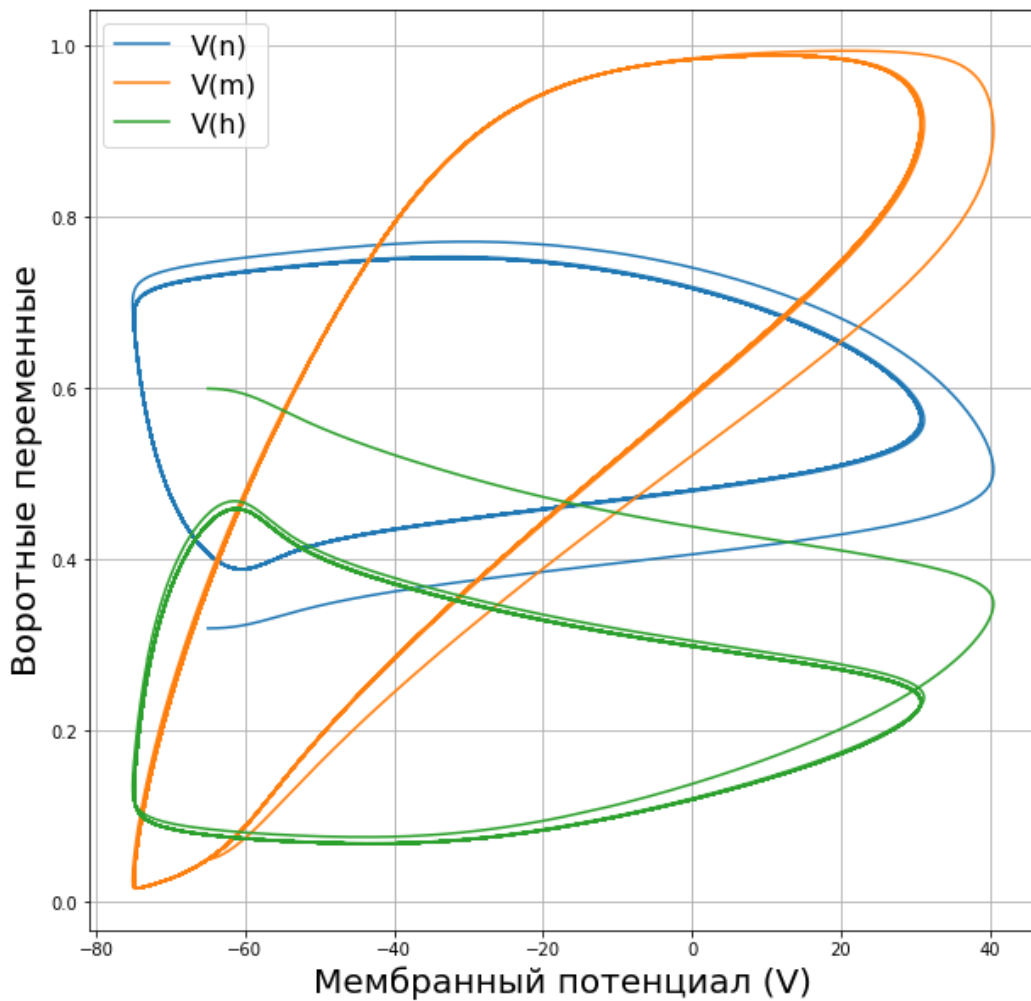


Рис. 6. Зависимость воротных переменных от мембранного потенциала V при значении входного тока $I_{ext} = 10$ мА.

Пусть теперь значение входного тока достигает значения $I_{ext} = 50$ мА. Если мы посмотрим на графики колебаний мембранного потенциала (рис. 7) и изменения воротных переменных, то увидим, что после первоначального скачка амплитуда колебания мембранного потенциала в периодическом режиме снижается. Это же происходит и с воротными переменными см. (рис. 8). Из рис. 9 следует, что выход на предельный цикл теперь происходит медленнее (траектория делает большее число витков).

При таком значении тока для достижения предельного цикла нужно сделать большее число витков по фазовой траектории. Такое значение входного тока губительно для нейронов головного мозга.

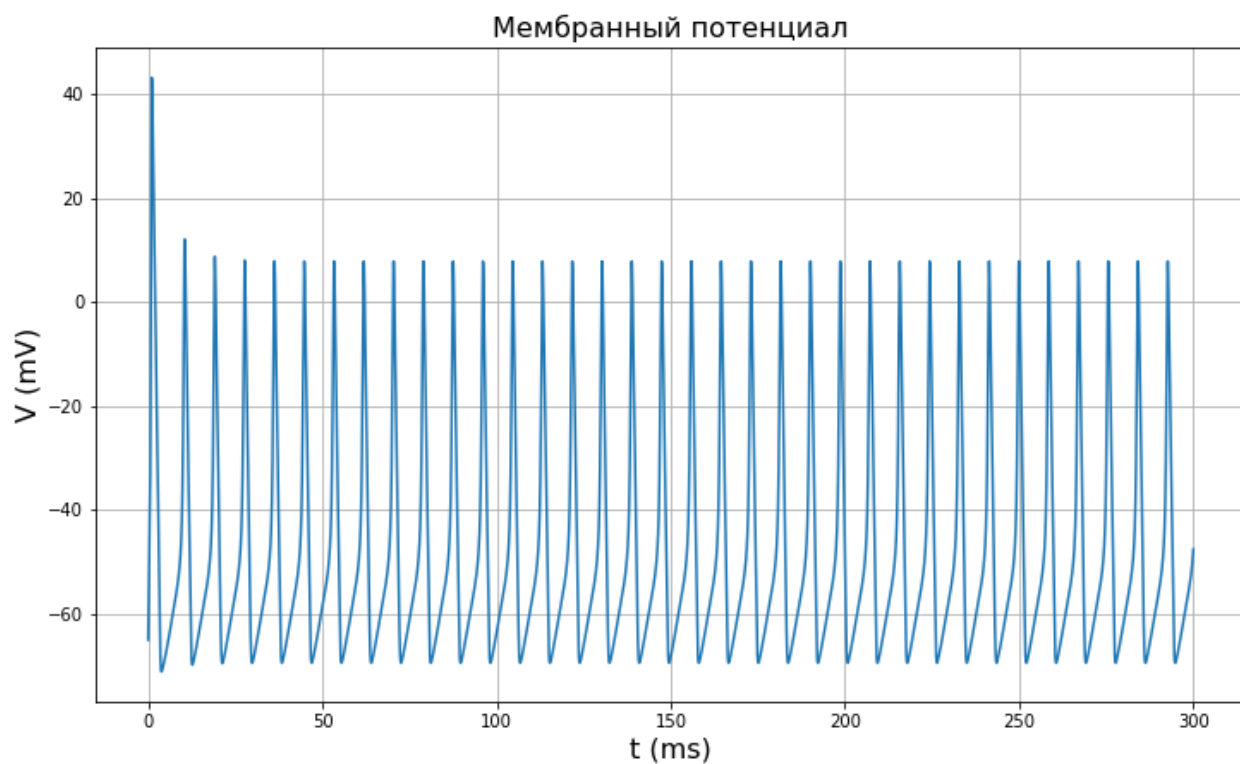


Рис. 7. Зависимость мембранного потенциала от времени по модели Ходжкина-Хаксли при значении входного тока $I_{ext} = 50$ мА.

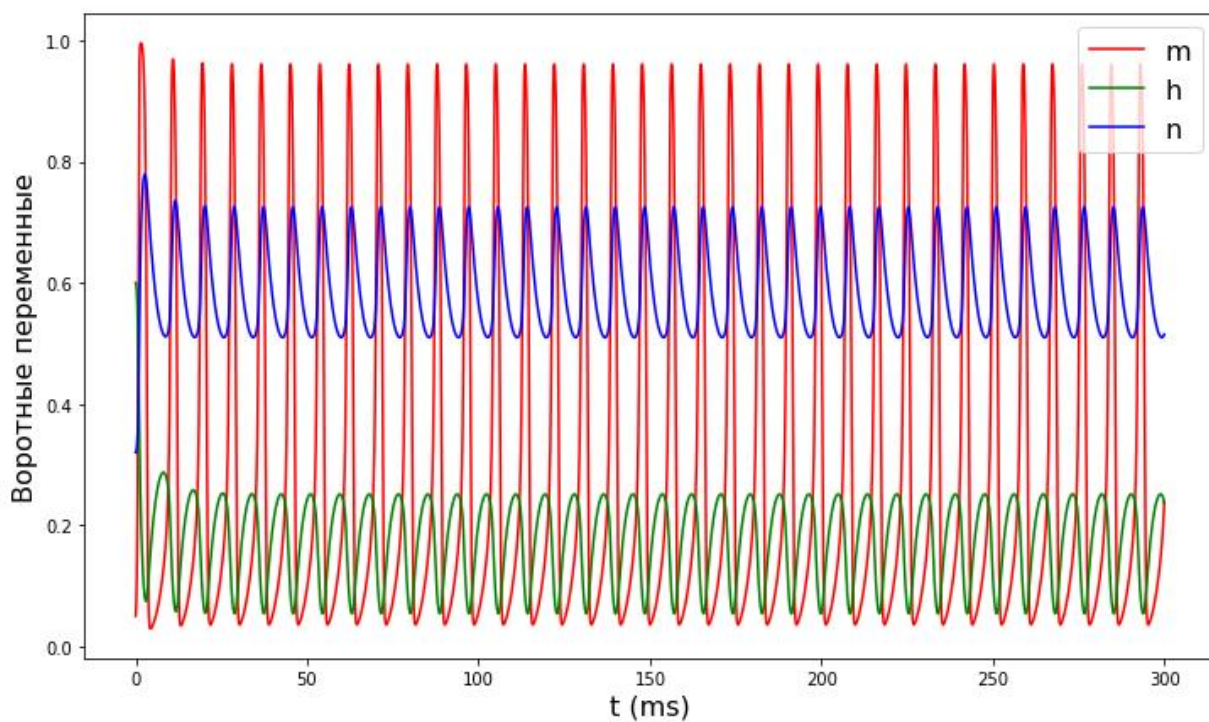


Рис. 8. Изменение воротных переменных с течением времени при значении входного тока $I_{ext} = 50$ мА.

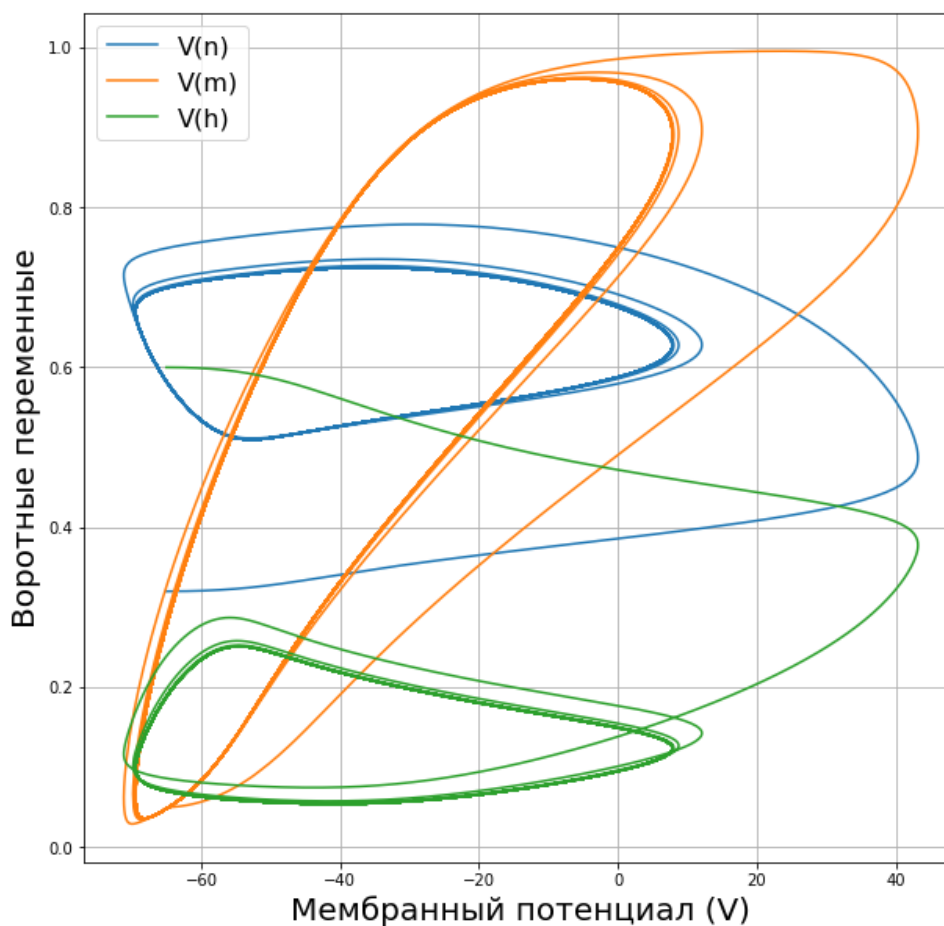


Рис. 9. Зависимость воротных переменных от мембранного потенциала V при значении входного тока $I_{ext} = 50$ мА.

Заключение. Таким образом, меняя значения входного тока в модели Ходжкина-Хаксли, можно сделать вывод, что наиболее оптимальным значением, при котором возможно возбуждение нейрона и передача сигналов другим нейронам, является значение входного тока, равное 10 мА.

СПИСОК ЛИТЕРАТУРЫ

1. Никифоров А. С., Гусев Е. И. Общая неврология: учебное пособие. – М.: ГЭОТАР-Медиа, 2007. – 720 с.
2. Шульговский В. В. Основы нейрофизиологии: учебное пособие для студентов вузов. – М.: Аспект Пресс, 2000. – 277 с.
3. Hodgkin A. L., Huxley A. F. Currents carried by sodium and potassium ions through the membrane of the giant axon of *Loligo* // Journal of Physiology. – 1952. – pp. 449–472.

ШАМАНАЕВ П. А., ГОЛОВАТЮК А. М.
ВЫЧИСЛЕНИЕ ПЕРИОДИЧЕСКИХ РЕШЕНИЙ НЕАВТОНОМНОГО
УРАВНЕНИЯ ДУФФИНГА МЕТОДОМ МАЛОГО ПАРАМЕТРА

Аннотация. В статье описан результат по автоматизации в математическом пакете Maple метода малого параметра для нахождения периодического решения неавтономного уравнения Дуффинга. Построены графики периодических решений и фазовые траектории уравнения Дуффинга при различных значениях малого параметра.

Ключевые слова: уравнение Дуффинга, периодические решения, резонанс, метод малого параметра, Maple.

SHAMANAEV P. A., GOLOVATYUK A. M.
CALCULATING PERIODIC SOLUTIONS TO NON-AUTONOMOUS
DUFFING EQUATION BY SMALL PARAMETER METHOD

Abstract. The article presents the result of automating the small parameter method in the Maple mathematical package for finding a periodic solution to the non-autonomous Duffing equation. Graphs of periodic solutions and phase trajectories of the Duffing equation are constructed for various values of a small parameter.

Keywords: Duffing equation, periodic solutions, resonance, small parameter method, Maple.

Введение. В 1918 году Г. Дуффинг ввел в рассмотрение уравнение с кубической нелинейностью для описания эффекта жесткости пружины [1]. В дальнейшем такого вида уравнения получили название уравнение Дуффинга. В работах [2; 3] изложены различные подходы к нахождению периодических решений неавтономного уравнения Дуффинга.

Одним из хорошо известных методов нахождения периодических решений нелинейных дифференциальных уравнений является метод малого параметра, предложенный А. Пуанкаре [4] и получивший свое развитие в последующих работах [5; 6].

В приведенных работах при вычислении периодического решения методом малого параметра, как правило, ограничиваются вычислениями первого или второго приближения в разложении искомого решения в ряд по малому параметру. В предлагаемой работе на основе автоматизации в математическом пакете Maple процесса нахождения приближений 2π -периодического решения неавтономного уравнения Дуффинга приведены результаты по вычислению приближений до третьего порядка включительно. Разработанный программный модуль в пакете Maple позволяет по наперед заданному произвольному числу приближений найти эти приближения.

Вычисление периодического решения неавтономного уравнения Дуффинга вдали

от резонанса. Рассмотрим уравнение Дуффинга в виде [5]

$$\ddot{x} + \omega^2 x = \lambda \sin t + \gamma \mu x^3, \quad (1)$$

где собственная частота ω – отличное от нечетного целого вещественное число, μ – малый параметр, λ и γ – некоторые вещественные параметры. Уравнение (1) будем называть возмущенным уравнением для случая вдали от резонанса.

Приведем результат работы программного модуля для $n = 3$, реализующего алгоритм вычисления методом малого параметра 2π -периодического решения неавтономного уравнения Дуффинга при условии, что значение собственной частоты находится вдали от резонансных значений.

Представим периодическое решение уравнения (1) в виде

$$x(t) = x_0(t) + \mu x_1(t) + \mu^2 x_2(t) + \mu^3 x_3(t) + o(\mu^3). \quad (2)$$

Подставляя выражение (2) в уравнение (1), приравнявая коэффициенты при одинаковых степенях малого параметра μ в левой и правой частях и пренебрегая слагаемыми порядка $o(\mu^3)$, получим систему уравнений

$$\begin{aligned} \ddot{x}_0 + \omega^2 x_0 &= \lambda \sin t, & \ddot{x}_1 + \omega^2 x_1 &= \gamma x_0^3, \\ \ddot{x}_2 + \omega^2 x_2 &= 3\gamma x_0^2 x_1, & \ddot{x}_3 + \omega^2 x_3 &= 3\gamma x_0^2 x_2 + 3\gamma x_0 x_1^2. \end{aligned} \quad (3)$$

Последовательно решая уравнения системы (3) в пакете Maple, получим

$$\begin{aligned} x_0(t) &= A_1 \sin t, & x_1(t) &= B_1 \sin t + B_3 \sin 3t, \\ x_2(t) &= C_1 \sin t + C_3 \sin 3t + C_5 \sin 5t, & x_3(t) &= D_1 \sin t + D_3 \sin 3t + \\ & & & + D_5 \sin 5t + D_7 \sin 7t, \end{aligned} \quad (4)$$

где

$$A_1 = \frac{\lambda}{\omega^2 - 1}, \quad B_1 = \frac{3\gamma\lambda^3}{4(\omega^2 - 1)^4}, \quad B_3 = -\frac{\gamma\lambda^3}{4(\omega^2 - 1)^3(\omega^2 - 9)},$$

$$C_1 = \frac{3\gamma^2\lambda^5(5\omega^2 - 41)}{8(\omega^2 - 1)^7(\omega^2 - 9)}, \quad C_3 = -\frac{3\gamma^2\lambda^5(5\omega^2 - 29)}{16(\omega^2 - 1)^6(\omega^2 - 9)^2},$$

$$C_5 = \frac{3\gamma^2\lambda^5}{16(\omega^2 - 1)^5(\omega^2 - 9)(\omega^2 - 25)},$$

$$\begin{aligned} D_1 &= \frac{3\gamma^3\lambda^7(35\omega^4 - 550\omega^2 + 2243)}{16(\omega^2 - 1)^{10}(\omega^2 - 9)^2}, D_3 \\ &= -\frac{9\gamma^3\lambda^7(7\omega^6 - 255\omega^4 + 2361\omega^2 - 6721)}{16(\omega^2 - 1)^{10}(\omega^2 - 9)^2(\omega^2 - 25)}, \end{aligned}$$

$$D_5 = \frac{3\gamma^3\lambda^7(7\omega^4 - 188\omega^2 + 901)}{16(\omega^2 - 1)^8(\omega^2 - 9)^2(\omega^2 - 25)^2},$$

$$D_7 = -\frac{3\gamma^3\lambda^7(\omega^2 - 13)}{16(\omega^2 - 1)^7(\omega^2 - 9)^2(\omega^2 - 25)(\omega^2 - 49)}.$$

Первое уравнение системы (3) будем называть невозмущенным уравнением для случая вдали от резонанса.

В качестве приближения порядка $o(\mu^3)$ к 2π -периодическому решению уравнения (1) возьмем

$$S_3(t) = x_0(t) + \mu x_1(t) + \mu^2 x_2(t) + \mu^3 x_3(t), \quad (5)$$

Приближения фазовых траекторий уравнения (1) будем строить как параметрические кривые $r(t) = (S_3(t), \dot{S}_3(t))$, $t \in [0, 2\pi]$.

Графики периодических решений и фазовые траектории уравнения Дуффинга вдали от резонанса. Поскольку в этом параграфе рассматривается не резонансный случай, то в качестве собственной частоты выберем, например, $\omega = 2$.

Приведем графики 2π -периодических решений и фазовых траекторий уравнения (1) при $\lambda = 1$ и $\gamma = 6$ и различных значениях параметра μ (рис. 1, 2).

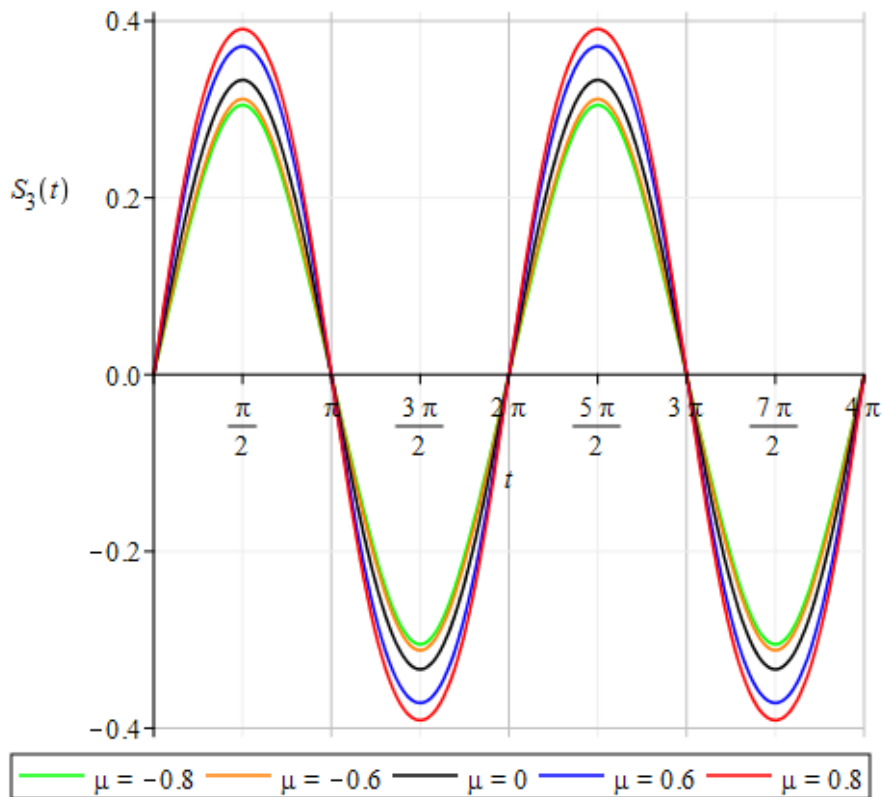


Рис. 1. Графики 2π - периодических решений уравнения (1) в случае отсутствия резонанса при различных значениях параметра μ .

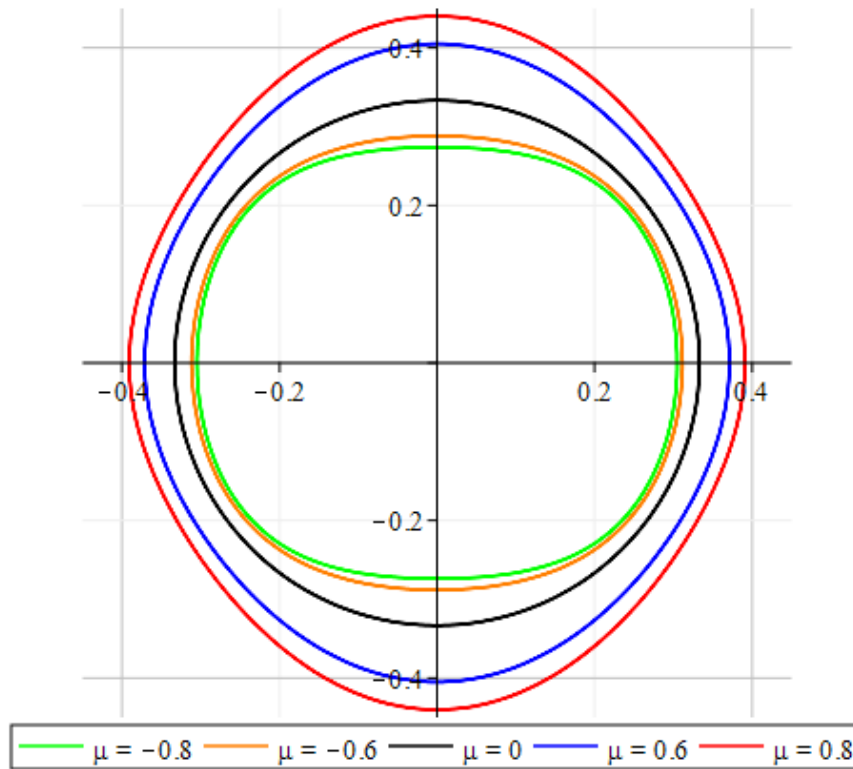


Рис. 2. Графики фазовых траекторий уравнения (1) в случае отсутствия резонанса при различных значениях параметра μ .

Вычисление периодического решения неавтономного уравнения Дуффинга при резонансе. Рассмотрим случай, когда собственная частота ω равна единице или меньше ее на некоторый малый параметр. Тогда согласно [5] представим собственную частоту ω в виде

$$\omega^2 = 1 + b\mu, \quad (6)$$

и рассмотрим два случая:

- 1) резонанс: $\omega^2 = 1, b = 0$;
- 2) вблизи резонанса: $\omega^2 = 1 + \mu, b = 1, \mu < 0$.

Тогда в первом случае уравнение (1) примет вид

$$\ddot{x} + x = \mu(\gamma x^3 + \lambda_1 \sin t), \quad (7)$$

во втором случае получим

$$\ddot{x} + x = \mu(-x + \gamma x^3 + \lambda_1 \sin t). \quad (8)$$

Здесь, $\lambda = \mu \lambda_1$. Уравнения (7) и (8) будем называть возмущенным уравнением для случаев 1 и 2, соответственно. Уравнение

$$\ddot{z} + z = 0,$$

– невозмущенным уравнением для случаев 1 и 2, соответственно.

В виду громоздкости формул, выражающих аналитическую зависимость приближений периодических решений от параметров γ и λ_1 , приведем результаты работы программного модуля для приближенного вычисления 2π -периодических решений с точностью $o(\mu^3)$ при значениях $\gamma = 6$ и $\lambda_1 = 1$ для случаев 1) $\omega^2 = 1$, $b = 0$

$$S_3(t) = -0,6057 \sin t + \mu(-0,1389 \cdot 10^{-1} \sin t - 0,4166 \cdot 10^{-1} \sin 3t) + \\ + \mu^2(0,6372 \cdot 10^{-2} \sin t + 0,1434 \cdot 10^{-1} \sin 3t - 0,2866 \cdot 10^{-2} \sin 5t) + \\ + \mu^3(-0,2679 \cdot 10^{-2} \sin t - 0,4470 \cdot 10^{-2} \sin 3t + 0,1447 \cdot 10^{-2} \sin 5t - \\ - 0,1971 \cdot 10^{-3} \sin 7t), \quad (9)$$

и случая 2) $\omega^2 = 1 + \mu$, $b = 1$, $\mu < 0$

$$S_3(t) = -0,7265 \sin t + \mu(-0,2789 \cdot 10^{-1} \sin t - 0,7189 \cdot 10^{-1} \sin 3t) + \\ + \mu^2(0,1448 \cdot 10^{-1} \sin t + 0,2542 \cdot 10^{-1} \sin 3t - 0,7117 \cdot 10^{-2} \sin 5t) + \\ + \mu^3(-0,6310 \cdot 10^{-2} \sin t - 0,6766 \cdot 10^{-2} \sin 3t + 0,3783 \cdot 10^{-2} \sin 5t - \\ - 0,7042 \cdot 10^{-3} \sin 7t). \quad (10)$$

Погрешность вычислений составляет 10^{-4} .

Графики периодических решений и фазовых траекторий уравнения Дуффинга при резонансе. Построим графики 2π -периодических решений и фазовых траекторий уравнения (1) для случаев 1) $\omega^2 = 1$, $b = 0$ (рис. 3, 4) и 2) $\omega^2 = 1 + \mu$, $b = 1$, $\mu < 0$ (рис. 5, 6) при различных значениях параметра μ .

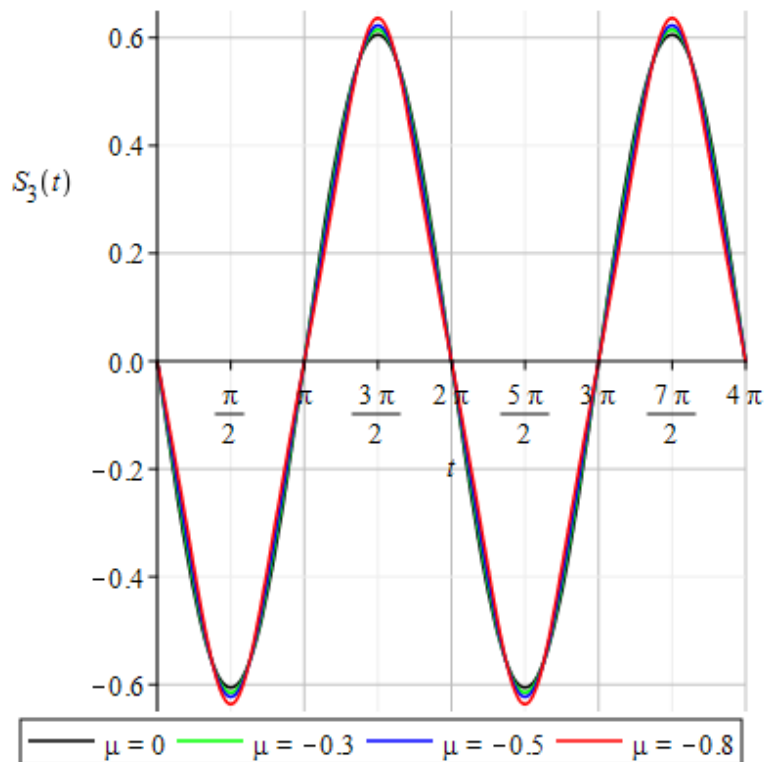


Рис. 3. Графики 2π - периодическому решению уравнения (7) в случае резонанса при $\omega^2 = 1$ и различных значениях параметра μ .

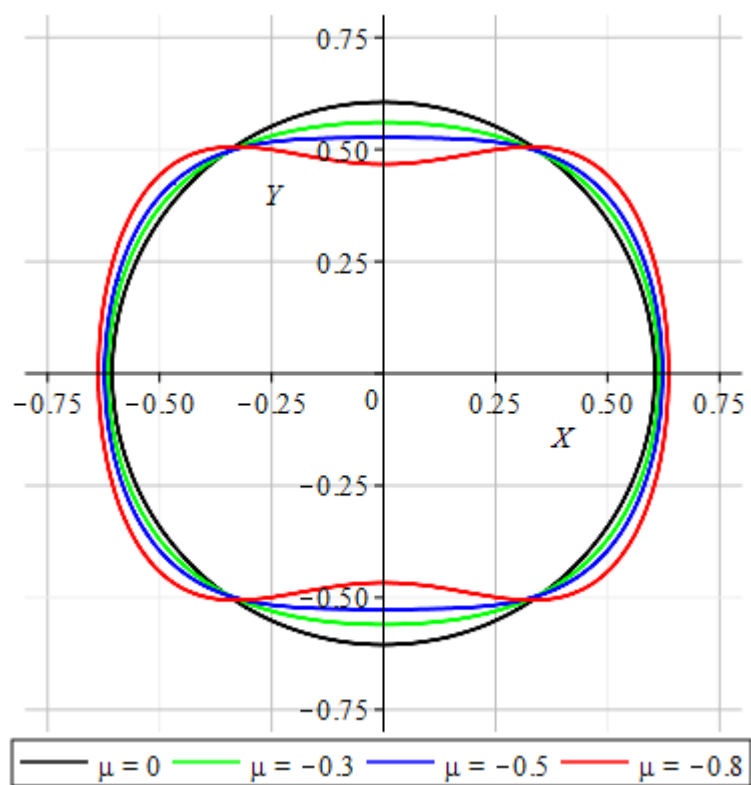


Рис. 4. Графики фазовых траекторий уравнения (7) в случае резонанса при $\omega^2 = 1$ и при различных значениях параметра μ .

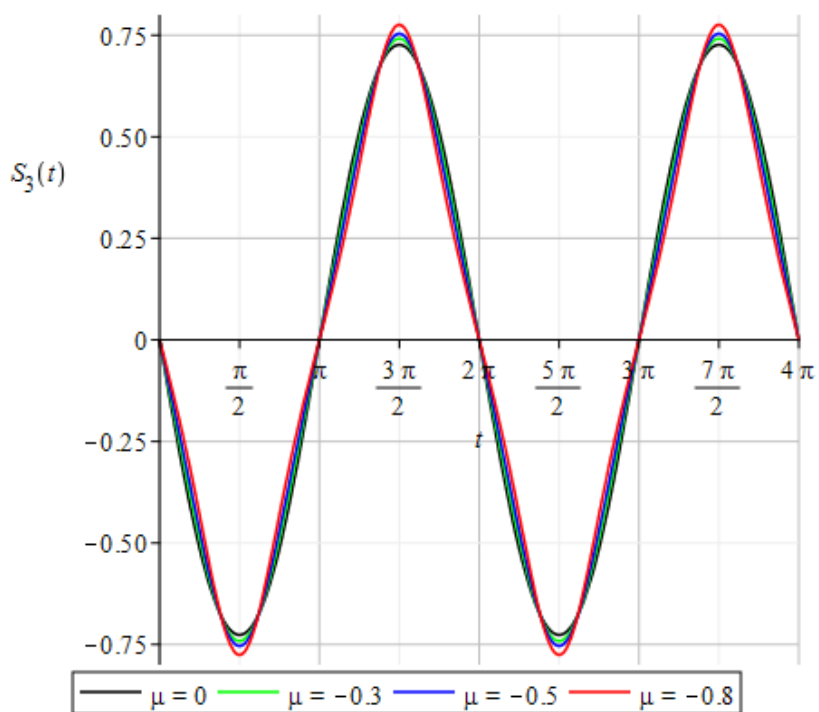


Рис. 5. Графики 2π - периодическому решению уравнения (7) вблизи резонанса при $\omega^2 = 1 + \mu$ и различных значениях параметра μ .

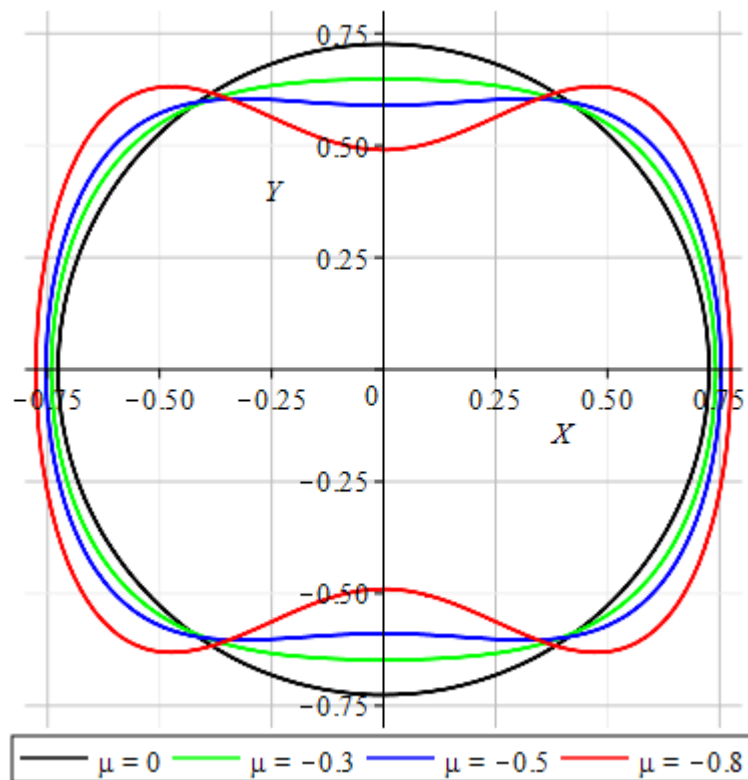


Рис. 6. Графики фазовых траекторий уравнения (7) вблизи резонанса при $\omega^2 = 1 + \mu$ и при различных значениях параметра μ .

Обсуждение полученных результатов. Достоверность результатов, полученных с помощью метода малых параметров, подтверждается сравнением их с численными решениями задачи Коши для соответствующих уравнений (1), (7) и (8). Начальные данные численных решений определялись из приближенных решений, полученных методом малого параметра.

Из графиков (рис. 1–6) видно, что случаи вдали резонанса и вблизи него имеют общую закономерность, заключающуюся в том, что при уменьшении параметра μ по модулю 2π -периодические решения и фазовые траектории возмущенных уравнений приближаются к 2π -периодическим решениям и фазовым траекториям соответствующих невозмущенных уравнений. Причем вдали резонанса амплитуда колебания 2π -периодических решений при положительных μ больше, чем при отрицательных.

Вместе с тем есть и существенные различия. Если в случае вдали от резонанса фазовые траектории представляют собой непересекающееся семейство замкнутых кривых, то в случае резонанса каждая фазовая траектория возмущенной системы пересекает фазовую траекторию невозмущенной системы в четырех точках. Причем в случае резонанса ($\omega^2 = 1$, $b = 0$) эти четыре точки пересечения для всех фазовых траекторий возмущенной системы остаются

неизменными, вблизи же резонанса ($\omega^2 = 1 + \mu$, $b = 1$, $\mu < 0$) соответствующие точки пересечения для различных фазовых траекторий возмущенной системы несколько смещаются.

СПИСОК ЛИТЕРАТУРЫ

1. Duffing G., *Erzwungene Schwingungen bei veranderlicher Eigenfrequenz und ihre technische Bedeutung*. – Braunschweig: Sammlung Vieweg, 1918. – 134 s.
2. Стокер Дж. *Нелинейные колебания в механических и электрических системах*. – М.: Издательство иностранной литературы, 1952. – 103 с.
3. Хаяси Т. *Нелинейные колебания в физических системах*. – М.: Мир, 1968. – 432 с.
4. Пуанкаре А. *Новые методы небесной механики. Избр. труды: Т. 1.* – М.: Наука, 1971. – 772 с.
5. Малкин И. Г. *Некоторые задачи теории нелинейных колебаний*. – М.: Наука, 1956. – 492 с.
6. Андронов А. А., Витт А. А., Хайкин С. Э. *Теория колебаний*. – М.: Гос. изд-во физико-мат. лит., 1959. – 916 с.

БУТКИНА А. А., КАДАНЦЕВ А. А., ШАМАЕВ А. В.
РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ УЧЕТА
И АНАЛИЗА ДИНАМИКИ ДВИЖЕНИЯ ДЕНЕЖНЫХ СРЕДСТВ
НА СЧЕТАХ ФИЗИЧЕСКОГО ЛИЦА

Аннотация. В статье описано разработанное авторами программное обеспечение, предназначенное для персонального учёта и анализа динамики движения денежных средств физического лица. Оно представляет собой веб-приложение, которое разработано на языках «JavaScript» и «PHP» и дополняет функционал существующих аналогичных систем.

Ключевые слова: веб-приложение, денежные средства, учёт финансов, СУБД MySQL, JavaScript, PHP, React, Redux.

BUTKINA A. A., KADANTSEV A. A., SHAMAEV A. V.
DEVELOPMENT OF SOFTWARE FOR ACCOUNTING AND ANALYSIS
OF DYNAMICS OF PERSONAL FUNDS MOVEMENT

Abstract. The article describes the software developed by the authors, designed for accounting and analysis of the dynamics of personal funds movement. It is a web application developed in JavaScript and PHP, expands and extends the functionality of existing analogues.

Keywords: web application, funds movement, personal accounting, MySQL DBMS, JavaScript, PHP, React, Redux.

Введение. Данное исследование посвящено разработке программного обеспечения для персонального учёта и анализа динамики движения денежных средств физического лица, являющегося обладателем банковских счетов и кредитных карт, которое расширяет и дополняет функционал существующих аналогичных систем. Актуальность исследования заключается в том, что разработанная в результате программная система позволит её пользователям сократить количество типичных ошибок при использовании кредитной карты, и тем самым минимизировать их финансовые риски при работе с кредитными картами, в особенности с картами, имеющими льготный период погашения задолженности.

Для достижения указанной цели были поставлены и решены следующие задачи:

- выполнить анализ предметной области;
- исследовать аналоги разрабатываемой программно-информационной системы;
- выполнить анализ и выбор инструментов разработки;
- выполнить концептуальное проектирование;
- выполнить проектирование интерфейса пользователя системы;
- реализовать программное обеспечение на выбранных языках программирования;

- выполнить проверку работоспособности созданного программного обеспечения.

Анализ предметной области. Первоочередным вопросом, вставшим перед авторами, являлся выбор формы реализации разрабатываемого программного обеспечения. В связи с высокой популярностью и распространенностью мобильных устройств, необходимо было сделать выбор между разработкой отдельного мобильного приложения и веб-приложения, способного работать на любом устройстве, на котором может быть запущен веб-браузер. В одном из проанализированных в работе источников [1] был проведен сравнительный анализ параметров их разработки, результаты которого представлены в таблице 1.

Таблица 1

Обзор параметров разработки веб и мобильного приложения

Параметры	Веб-приложение	Мобильное приложение
Масштаб	Возможность одновременного использования большим количеством пользователей	7-10 человек
Зависимость от ОС	Работает на любой платформе, где можно быть запущен веб-браузер	Создаются на базе ОС Android, iOS. Для создания приложения используются разные языки программирования
Зависимость от сети	Для работы необходим веб-браузер и наличие доступа в сеть Интернет, что может потребовать дополнительных затрат. Не работает в режиме офлайн. Необходимо следить за расходуемым трафиком	Не зависит от сети (если только не требуется подключить дополнительные возможности). Работает в режиме офлайн после установки приложения
Доступ к хранилищу устройств	Требуют наличия определенных прав доступа к хранилищу при установке	Обеспечивают доступ
Привлечение аудитории	Чтобы веб-приложение было успешным, часто необходимо вкладывать значительные средства в его продвижение.	Достаточно разработать удобное приложение, чтобы пользователи магазина приложений могли выбрать его из числа доступных. Устанавливается постоянный контакт с пользователями с помощью рассылок, сообщений о скидках и акциях. Пользователи могут оставить негативные отзывы о работе приложения, что повлияет на выбор других пользователей
Память	Не требуют установки на мобильном устройстве пользователя	Устанавливается на мобильное устройство пользователя непосредственно из магазина приложений (Google Play, App Store), требует наличия определенного объема свободной памяти на устройстве
Функционал	Простой, удобный. Меньше времени и затрат на разработку	Сложный, с множеством функций, удобный в использовании, более качественный для пользователя (при схожем функционале). Больше времени и затрат на разработку. Является дополнением к веб-приложению
Обновление	Обновление приложения происходит автоматически на сайте и доступно пользователям в любое время	При изменении дизайна, внесении новых функций, новую версию нужно заново скачать из магазина приложений
Оплата	Необходимость внесения ежемесячной оплаты за работу с приложением или, в зависимости от объема доступных операций. Приложение может предоставляться бесплатно, а доступ к дополнительным премиум-функциям может быть предоставлен за отдельную плату. Разработка обходится дешевле	Меньше расходов на продвижение приложения в связи с тем, что магазины приложений посещают целевые пользователи, которые могут ознакомиться с отзывами и выбрать оптимальное приложение. Отсутствуют затраты на поисковую оптимизацию

На основе анализа приведенной в таблице 1 информации был сделан вывод о том, что в рамках данного исследования будет выполнена разработка программного обеспечения в форме веб-приложения, так как оно обладает следующими преимуществами:

- Отсутствие требований к операционным системам пользователей. Веб-приложения нетребовательны к ресурсам и не предъявляют никаких требований к аппаратной платформе;
- Возможность адаптации интерфейса приложения к работе на разных устройствах. Инструменты разработки клиентской части веб-приложений позволяют адаптировать интерфейс для его корректного отображения на разных типах устройств;
- Отсутствие необходимости выполнять обновления на стороне пользователя. Обновление приложения происходит автоматически;
- Требуется меньший объем ресурсов для разработки.

Анализ аналогов. Разрабатываемое веб-приложение позволит его пользователям выполнять следующие действия: контролировать свои денежные потоки, прогнозировать расходы, воздерживаться от чрезмерных трат, наглядно отслеживать, на что были потрачены и откуда поступили денежные средства, делать накопления, и как следствие, улучшать своё финансовое состояние. Поэтому у разрабатываемой системы существуют следующие известные аналоги: «CoinKeeper» [2], «Дзен мани» [3] и «Money Lover» [4]. Основные результаты проведенного авторами сравнительного анализа перечисленных выше систем приведены в таблице 2.

Таблица 2

Сравнение рассматриваемых аналогов системы

Функция	CoinKeeper	Дзен Мани	Money Lover
Интерфейс			
Локализация	Полная	Полная	Частичная
Безопасность			
Многофакторная авторизация	–	–	–
Резервное копирование в нескольких местах	–	–	–
Хранение информации на собственном сервере	+	+	+
Личные финансы			
Условия использования веб-версии	Условно- бесплатное	Условно- бесплатное	Только платная версия
Отображение данных в графической форме	+	+	+
Анализ расходов	+	+	+
Отчёты	Некоторые отчеты доступны только в платной версии	Некоторые отчеты доступны только в платной версии	+
Поддержка валют	+	+	+
Постановка целей	–	–	–
Планирование финансов	В платной версии	+	+
Поддержка электронных систем расчётов	+	–	–
Повторы операций	+	+	–
Напоминания о платежах	+	+	+
Экспорт / импорт данных	В платной версии	+	В платной версии

Контроль и ограничение расходов	+	+	–
Возможность прикреплять заметки	+	+	–
Напоминание о долгах	+	+	–

Таким образом, анализ существующих аналогов показывает, что абсолютно все рассмотренные веб-приложения содержат следующие возможности:

- 1) семейный доступ;
- 2) поддержка нескольких валют;
- 3) упоминания и заметки.

Веб-приложение «CoinKeeper» предусматривает полнофункциональную платную версию, которая реализует все преимущества данной системы. Веб-приложение «Дзен мани» воплотило в себе все достоинства, кроме возможности прикрепления заметок. Веб-приложение «Money Lover» имеет множество недостатков и значительно уступает остальным приложениям по функционалу.

Многие разработчики приложений для персонального учета финансов уделяют больше внимания дизайну приложения, забывая о его функциональных возможностях. Первое впечатление о приложении у большинства его пользователей складывается на основе его дизайна, однако как показывает практика использования приложений, впоследствии они обнаруживают, что выбранное ими приложение не оправдывает их ожиданий с точки зрения реализуемого функционала и перестают им пользоваться.

Поэтому хорошее приложение для персонального учета и анализа финансов (равно как и любое другое) требует соблюдения баланса между дизайном и функционалом.

Технологии реализации. Так как разрабатываемое приложение включает клиентскую (frontend) и серверные (backend) части, далее опишем инструменты, применяемые при разработке этих частей.

При разработке frontend части веб-приложения используются:

- стандартный язык разметки документов в Интернет HTML;
- каскадные таблицы стилей (CSS);
- JavaScript-библиотека React для создания пользовательских интерфейсов;
- JavaScript библиотека Redux для управления состоянием приложения;
- Chart.JS – JavaScript-библиотека визуализации данных;
- Toastify – библиотека для отображения уведомлений с настраиваемым текстом.

При разработке backend части веб-приложения используются скриптовый язык программирования PHP и система управления базами данных (СУБД) MySQL.

Также в ходе исследования применялись:

- методология Unified Process;
- архитектурный паттерн MVC;
- архитектурный стиль взаимодействия компонентов распределенного приложения (REST).

Проектирование. Эффективность любого приложения, использующего в своей работе базу данных, зависит от того, насколько оптимально спроектирована ее схема. Она напрямую влияет на такие характеристики, как общая архитектура системы, набор классов, скорость обработки данных и удобство обращения к хранилищу.

Поскольку создаваемое веб–приложение предназначено для осуществления персонального учета и анализа финансов его пользователей, то используемая в нём база данных (БД) должна хранить информацию о счетах и транзакциях каждого пользователя. Чтобы реализовать такую БД, необходимо определить входные и выходные данные, которыми должна оперировать система. Для начала определим, что будем понимать под термином «входная информация». Входная информация – это данные, которые поступают в систему извне. Для исследуемого бизнес–процесса в качестве входных данных выступает приведенный ниже набор вводимых пользователем сведений:

- Логин, в качестве которого используется адрес электронной почты, и пароль для авторизации в системе;
- Типы счетов пользователя: счета для учета имеющихся у пользователя на руках наличных средств; банковские (в том числе депозитные, накопительные и прочие) счета; счета кредитных или дебетовых карт;
- Информация о счете: банк, в котором открыт счет; процентная ставка кредитования; процент на остаток для счетов с накоплением.
- Данные по картам с беспроцентным периодом: максимальная продолжительность беспроцентного периода; продолжительность периода отсутствия выдачи уведомлений о досрочном погашении задолженности в рамках беспроцентного периода; размер лимита кредитования; размер определяемого пользователем лимита на объём используемых кредитных средств для выдачи предупреждения о его превышении.
- Первоначальный объём средств на счетах пользователя;
- История транзакций;
- Категория транзакции (при её добавлении);
- Комментарии к транзакциям.

В процессе разработки веб–приложения была спроектирована схема базы данных, состоящая из 11 таблиц (рис. 1). Теперь опишем взаимосвязи между таблицами, которые продемонстрированы на **Ошибка! Источник ссылки не найден.. Ошибка! Источник**

ссылки не найден. «User» связана отношением «один-ко-многим» по полю «userID» с **Ошибка! Источник ссылки не найден.** «Cash», «Card», «CreditCard», «bankAccount», «deposit» и «transaction». **Ошибка! Источник ссылки не найден.** «CreditCard» связана отношением «один-к-одному» по полю «paymentID» с **Ошибка! Источник ссылки не найден.** «paymentCreditCard». **Ошибка! Источник ссылки не найден.** «transaction» связана с **Ошибка! Источник ссылки не найден.** «comments» отношением «один-к-одному» по полю «commentID».

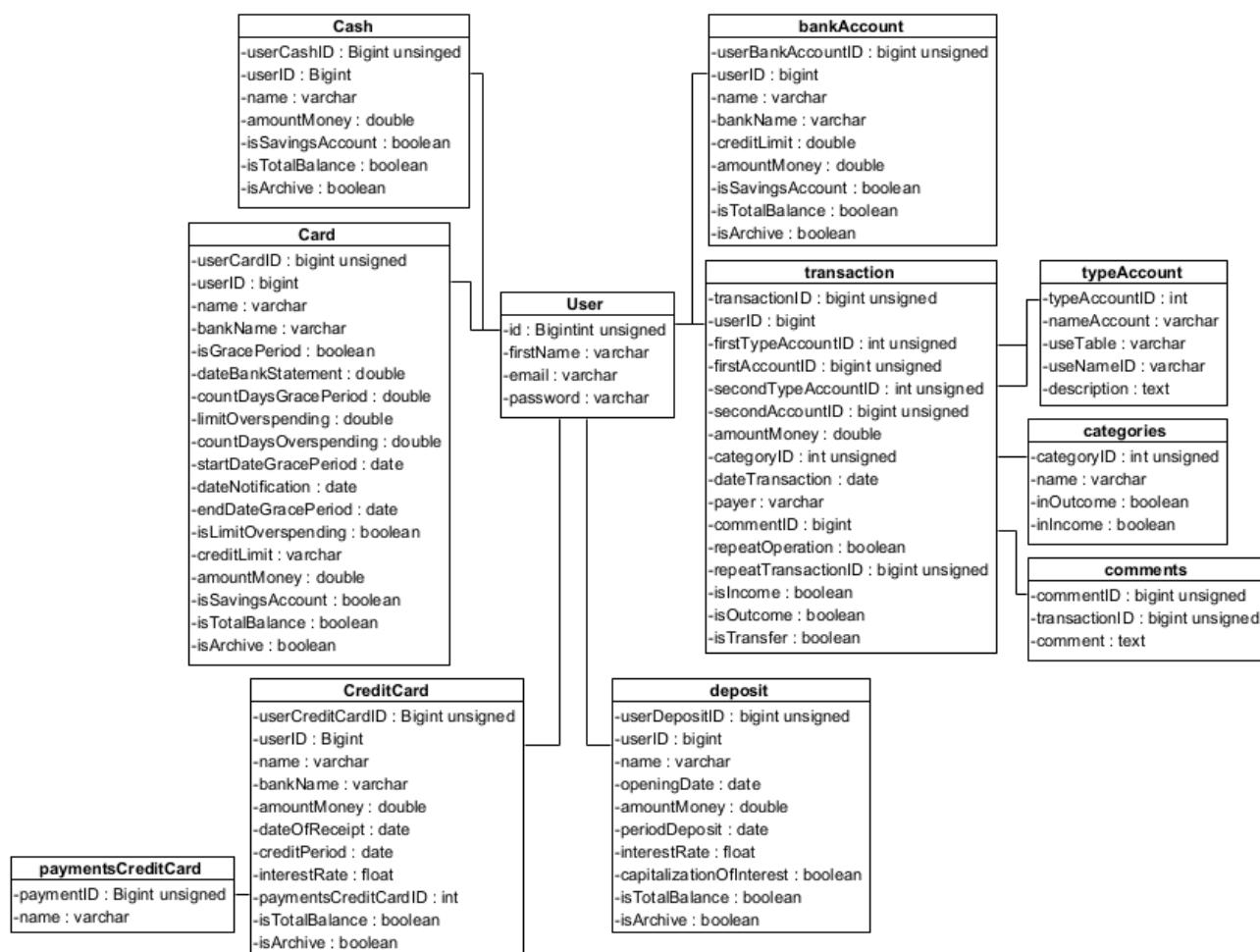


Рис. 1. Схема БД разработанного веб-приложения.

Выходная (результативная) информация – это информация, генерируемая на выходе системы. Основной выходной информацией в разрабатываемой системе являются следующие списки и отчёты: «Список счетов пользователя», «Список транзакций пользователя» и «Отчет о текущих расходах и доходах за месяц».

Система должна отображать информацию о списке счетов пользователя, сформированную на основе **Ошибка! Источник ссылки не найден.** «Cash», «Card», «CreditCard», «bankAccount» и «deposit». Система должна отображать информацию о списке транзакций пользователя на основе **Ошибка! Источник ссылки не найден.** «transaction».

Отчет о текущих расходах и доходах за месяц должен показывать избыточные расходы (перерасход) пользователя на основе анализа соотношения сумм всех его расходов и доходов за этот период.

Реализация приложения. Опишем алгоритм работы системы уведомлений пользователя, реализованный в разработанной системе, с помощью диаграммы деятельности (рис. 2). Для того чтобы пользователь системы, являющийся владельцем кредитной карты с беспроцентным периодом, смог получать уведомления о превышении его бюджета и приближении даты окончания указанного периода, он должен при создании или редактировании параметров счета карты установить флажок в поле «Беспроцентный период», что приведет к необходимости заполнения совокупности следующих полей для ввода числовых данных:

- сумма расходов, которую пользователь не хотел бы превысить (необязательно);
- максимальная длительность беспроцентного периода (в днях);
- день месяца, в который происходит формирование банковской выписки о состоянии текущей задолженности (целое число, от 1 до 31);
- количество дней, остающихся до окончания срока действия текущего беспроцентного периода (по умолчанию – 3 дня).

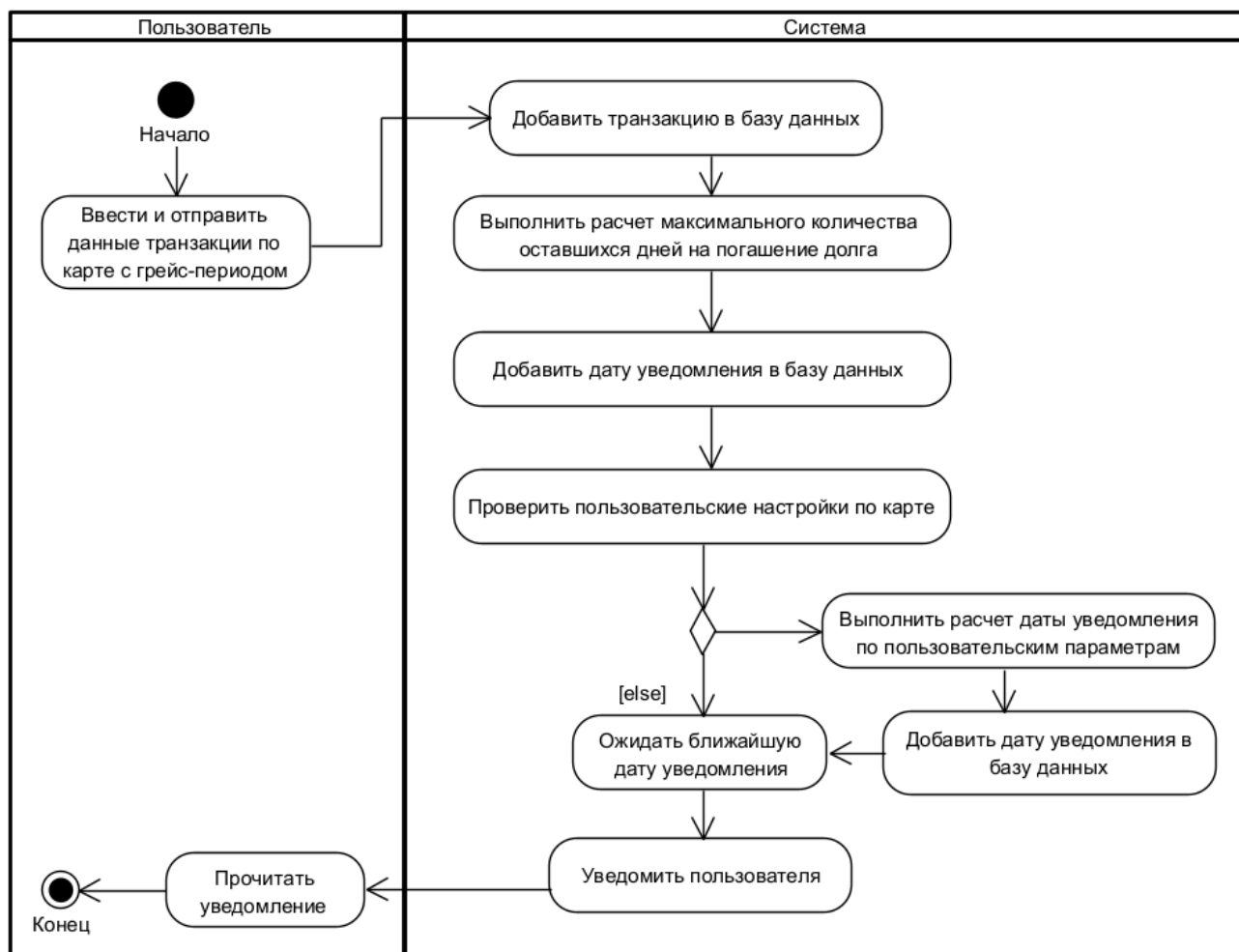


Рис. 2. Диаграмма деятельности.

Запуск алгоритма работы системы уведомлений пользователю происходит при совершении пользователем расходной транзакции по данному счету. В случае, если при совершении очередной (расходной) транзакции в систему сумма на счете станет отрицательной, система должна будет автоматически произвести расчет ближайшей даты выдачи уведомления и даты окончания беспроцентного периода по карте на основе указанных выше данных и выполнить добавление этой информации в БД. Если же в результате совершения очередной (приходной) транзакции сумма на счете вновь станет положительной, выдача уведомлений будет отменена до следующего момента получения отрицательного значения суммы на счете. Таким образом, при наступлении ближайшей даты, в которую система должна отправить уведомление, на экране пользователя при его входе в систему должно отобразиться окно с соответствующим сообщением (рис. 3), содержимое которого также будет продублировано в тексте письма, отправленного на его электронную почту в этот же день.

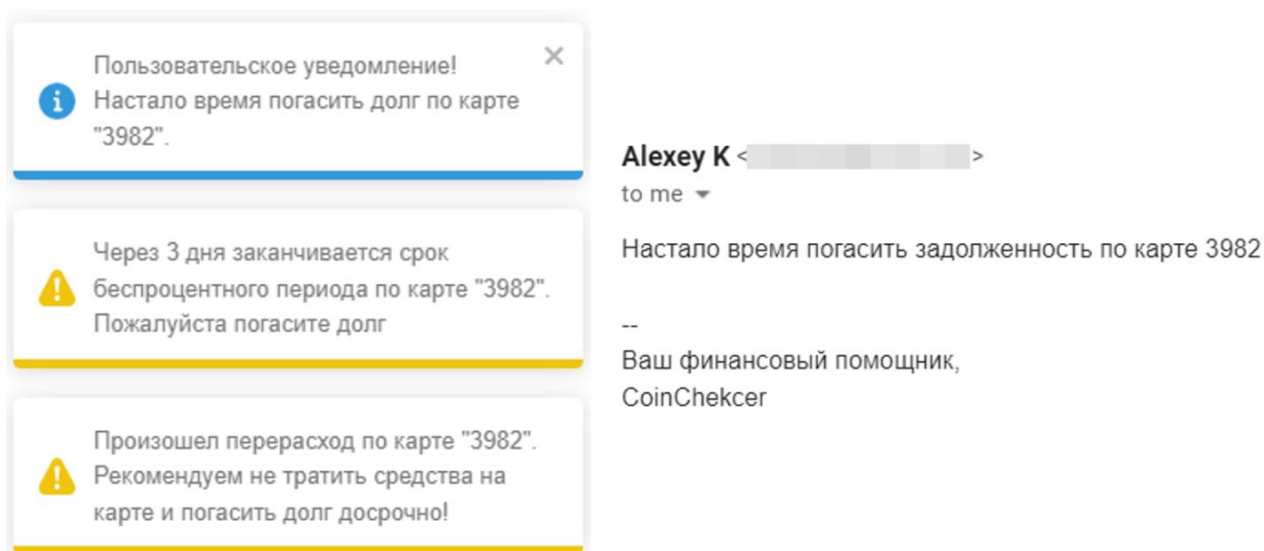


Рис. 3. Примеры уведомлений пользователя.

После успешной авторизации пользователя в системе, он должен будет добавить хотя бы один счет на странице «Счета» (рис. 4). После добавление счетов, участвующих в персональном денежном обороте, пользователь может перейти к добавлению транзакций. Для этого ему необходимо перейти на страницу «Транзакции» (рис. 5) и нажать на кнопку, соответствующую виду транзакции: «расход», «доход» или «перевод». Также в системе пользователь имеет возможность просмотреть график, отражающий изменение баланса его средств за последние 30 дней, а также распределение расходов по категориям и отчет по движению средств за текущий месяц с помощью страницы «Обзор» (рис. 6).

Также, в разработанном веб-приложении пользователям доступен персональный советник по управлению финансами, который находится на странице «Советник». В частности, в текущей версии помощника реализован инструмент, выполняющий анализ расходов пользователей, результатом которого является получение средних значений расходов в каждой из предусмотренных в приложении категорий за последние полгода. Результатом данного анализа является выдача пользователю индивидуальных рекомендаций по формированию финансового резерва на случай возникновения непредвиденных жизненных ситуаций, иначе называемого «финансовой подушкой». Пример того, как может выглядеть подобная рекомендация, приведен на рисунке 7.

Обзор	Транзакции	Советник (т)	Счета	Выйти
Наличные				
Наличный счёт 1		1 000 руб.	Изменить	Удалить
Наличный счёт 2		300 000 руб.	Изменить	Удалить
Карты				
Карта 1		15 000 руб.	Изменить	Удалить
Grace		-20 руб.	Изменить	Удалить
Кредиты				
Кредит		1 руб.	Изменить	Удалить
Банковские счета				
Банковский счёт		1 000 000 руб.	Изменить	Удалить
Вклады				
Вклад 1		100 000 руб.	Изменить	Удалить

Всего средств
1 415 981 руб.

Добавить

Наличный счет
Карту
Кредит

Банковский счёт
Вклад

Рис. 4. Страница «Счета».

Обзор	Транзакции	Советник (т)	Счета	Выйти
Наличные транзакции				
Добавить одну транзакцию				
15.7.2022	Перевод со счета "Наличный счёт 1" на счет "Наличный счёт 2"	10 000 руб.	Наличный счёт 1 -> Наличный счёт 2	Всего средств 1 415 981 руб.
	Редактировать Удалить			
11.7.2022	Образование - Курсы 1С	6 000 руб.	Наличный счёт 1	Наличные
	Покупка курсов на сайте			Наличный счёт 1 1 000 руб.
	Редактировать Удалить			Наличный счёт 2 300 000 руб.
10.7.2022	Зарплата - Фриланс	5 000 руб.	Наличный счёт 1	Карты
	Пополнение счета			Карта 1 15 000 руб.
	Редактировать Удалить			Grace -20 руб.
5.5.2022	Зарплата - МГУ им. Н.П. Огарева	4 000 руб.	Наличный счёт 1	Кредиты
	Пополнение счета			Кредит 1 руб.
	Редактировать Удалить			Банковские счёта
2.5.2022	Машина - АЗС	2 000 руб.	Наличный счёт 1	Банковский счёт 1 000 000 руб.
	АЗС/АИ-95			Вклады
	Редактировать Удалить			Вклад 1 100 000 руб.
1.5.2022	Кафе и рестораны -	1 000 руб.	Наличный счёт 1	
	Редактировать Удалить			
15.4.2022	Перевод со счета "Наличный счёт 1" на счет "Наличный счёт 2"	10 000 руб.	Наличный счёт 1 -> Наличный счёт 2	
	Редактировать Удалить			
15.4.2022	Перевод со счета "Наличный счёт 1" на счет "Наличный счёт 2"	10 000 руб.	Наличный счёт 1 -> Наличный счёт 2	
	Редактировать Удалить			

Рис. 5. Страница «Транзакции».

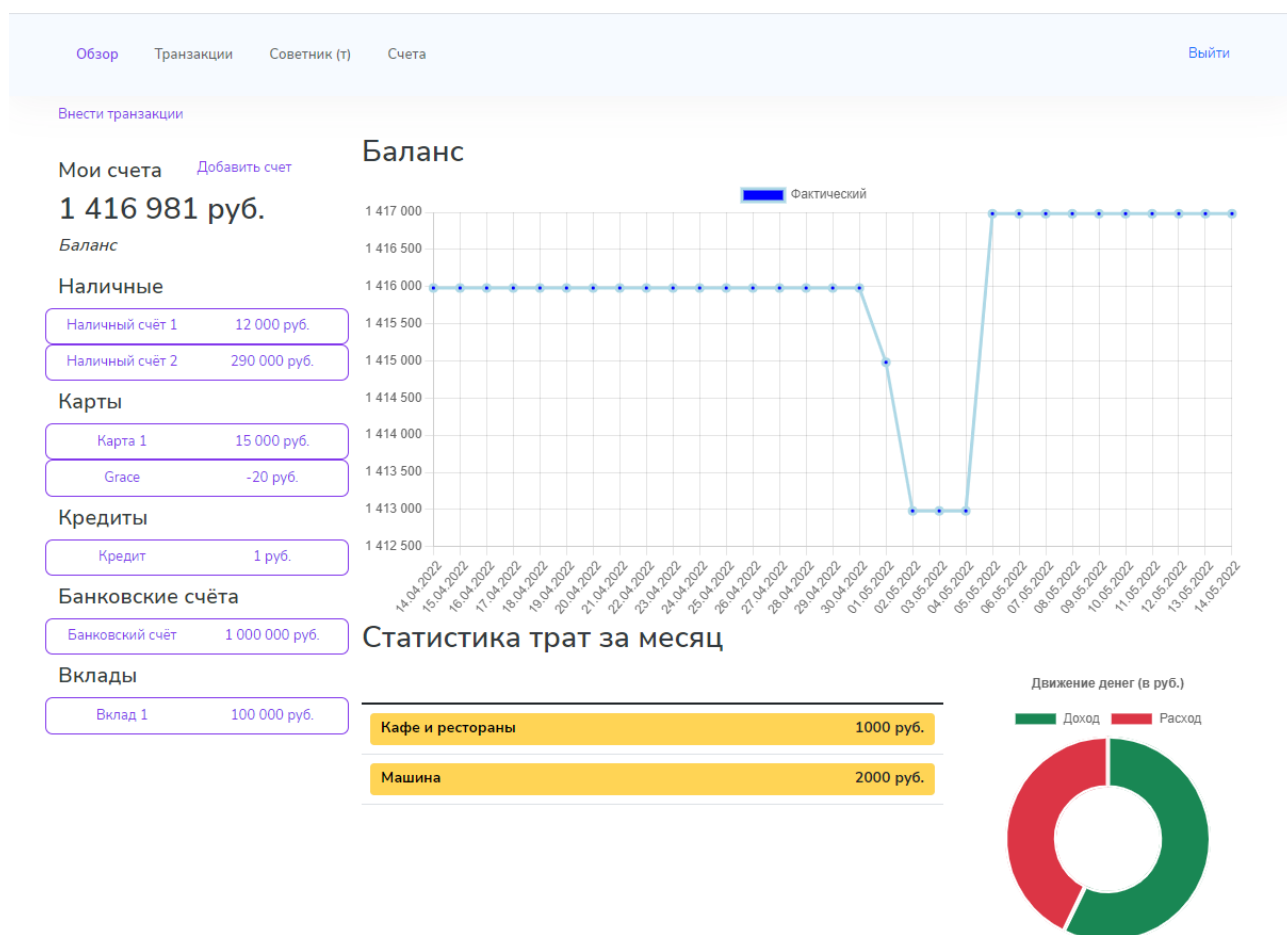


Рис. 6. Страница «Обзор».

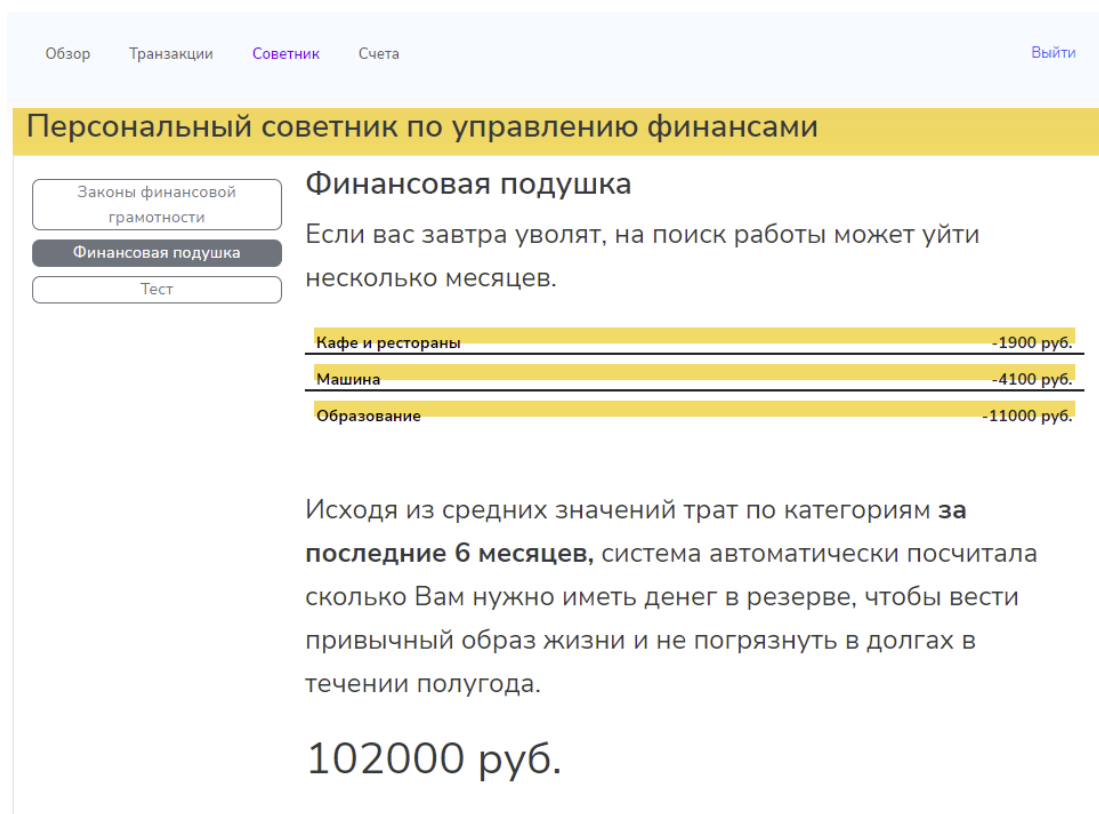


Рис. 7. Страница «Советник».

Заключение. Все задачи, поставленные в рамках данного исследования, были решены. Результатом работы является разработанное веб-приложение, предназначенное для персонального учёта и анализа динамики движения денежных средств физического лица, функционал которого, в отличие от аналогичных систем, помимо стандартных включает следующие возможности:

- отправка пользователю системы, являющемуся владельцем кредитной карты с беспроцентным периодом, уведомлений о превышении его бюджета и приближении даты окончания указанного периода;
- наличие помощника по управлению финансовыми активами пользователя.

Практическая ценность работы состоит в том, что разработанное в результате её выполнения веб-приложение позволит его пользователям минимизировать финансовые риски при работе с кредитными картами, в том числе с картами, имеющими льготный период погашения задолженности.

СПИСОК ЛИТЕРАТУРЫ

1. Панченко Н. В. Особенности разработки веб-приложений и мобильных приложений // Тенденции развития науки и образования. – 2019. – № 57-2. – С. 24–26.
2. CoinKeeper: учет расходов и доходов, бюджет семьи, личные финансы [Электронный ресурс]. – Режим доступа: <https://about.coinkeeper.me/> (дата обращения: 09.04.2022).
3. Дзен-мани: онлайн управление личными деньгами, планирование и учет личных финансов [Электронный ресурс]. – 2010. – Режим доступа: <https://zenmoney.ru/> (дата обращения: 09.04.2022).
4. Money Lover: No.1 Expense Manager Budget Planner [Электронный ресурс]. – Режим доступа: <https://moneylover.me/> (дата обращения: 09.04.2022).

БЕРСЕНЕВ К. С., ФИРСОВА С. А.

**РАЗРАБОТКА И РЕАЛИЗАЦИЯ МЕТОДИКИ РАСЧЕТА ТЕХНОЛОГИЧЕСКОЙ
ОСНАСТКИ ДЛЯ ИЗГОТОВЛЕНИЯ ОТДЕЛЬНЫХ УЗЛОВ
ГРУЗОВЫХ ВАГОНОВ ПРИ ИХ ПРОЕКТИРОВАНИИ**

Аннотация. В статье рассматривается процесс разработки методики расчета технологической оснастки для изготовления отдельных узлов грузовых вагонов. Разработанная методика была реализована с помощью созданной авторами программно-информационной системы, в основе которой лежит применение программных комплексов ANSYS 2022 и КОМПАС-3D, а также СУБД Microsoft SQL Server.

Ключевые слова: технологическая оснастка, методика, проектирование, КОМПАС-3D, ANSYS, прочностной расчет, информационная система, Visual Studio, C#, разработка, Microsoft SQL Server.

BERSENEV K. S., FIRSOVA S. A.

**DEVELOPMENT AND IMPLEMENTATION OF METHODOLOGY FOR
CALCULATING TECHNOLOGICAL EQUIPMENT FOR MANUFACTURING OF
INDIVIDUAL UNITS OF FREIGHT CARS DURING THEIR DESIGN**

Abstract. The article discusses the process of developing a methodology for calculating technological equipment for manufacturing of individual units of freight cars. The developed methodology is implemented using a software and information system created by the authors, which is based on the use of ANSYS 2022 and Compass-3D software complexes, as well as Microsoft SQL Server DBMS.

Keywords: technological equipment, methodology, design, COMPASS-3D, Ansys, strength calculation, information system, Visual Studio, C#, development, Microsoft SQL Server.

В настоящее время в таких отраслях тяжелой промышленности, как авиастроение, автомобилестроение, машиностроение, и в частности, вагоностроение широко применяют CAD/CAM/CAE системы. Например, на предприятии RM-Rail Рузхимаш при проектировании грузовых вагонов используют систему 3D-моделирования КОМПАС-3D. Одним из основных этапов проектирования вагонов является прочностной расчет их металлоконструкций. Однако именно этот этап на предприятии не автоматизирован. Поэтому было решено разработать методику автоматизированного расчета технологической оснастки, используемой для изготовления грузовых вагонов, с помощью современных CAD и CAE – систем.

Технологическая оснастка представляет собой вспомогательное устройство, предназначенное для реализации технологических возможностей оборудования, или автономно работающее устройство, в котором использованы ручные, пневматические, электромеханические и другие приводы [1]. Конструкцию оснастки выбирают с учетом стандартных и типовых решений для данного вида технологической операции с учетом габаритных размеров изделия, вида заготовки, характеристики материала изделия, точности параметров изделия, технологических схем базирования и фиксации изделий, характеристик оборудования, типа производства [2].

Проектирование технологической оснастки состоит из следующих этапов:

- создание трехмерной модели;
- выполнение прочностного расчета металлоконструкции;
- выбор сечения профиля на основе максимального допустимого значения прогиба конструкции;
- разработка документации для изготовления технологической оснастки.

Однако, следует отметить, что процесс ручного расчета достаточно сложный и трудоемкий, для экономии времени применяют грубый расчет, в результате которого проектируемое изделие точно отвечает необходимым прочностным характеристикам, что ведет к лишним материально финансовым затратам.

В связи с этим предложим методику, способствующую более быстрому и точному расчету металлоконструкций, представив ее в виде блок-схемы на рисунке 1.

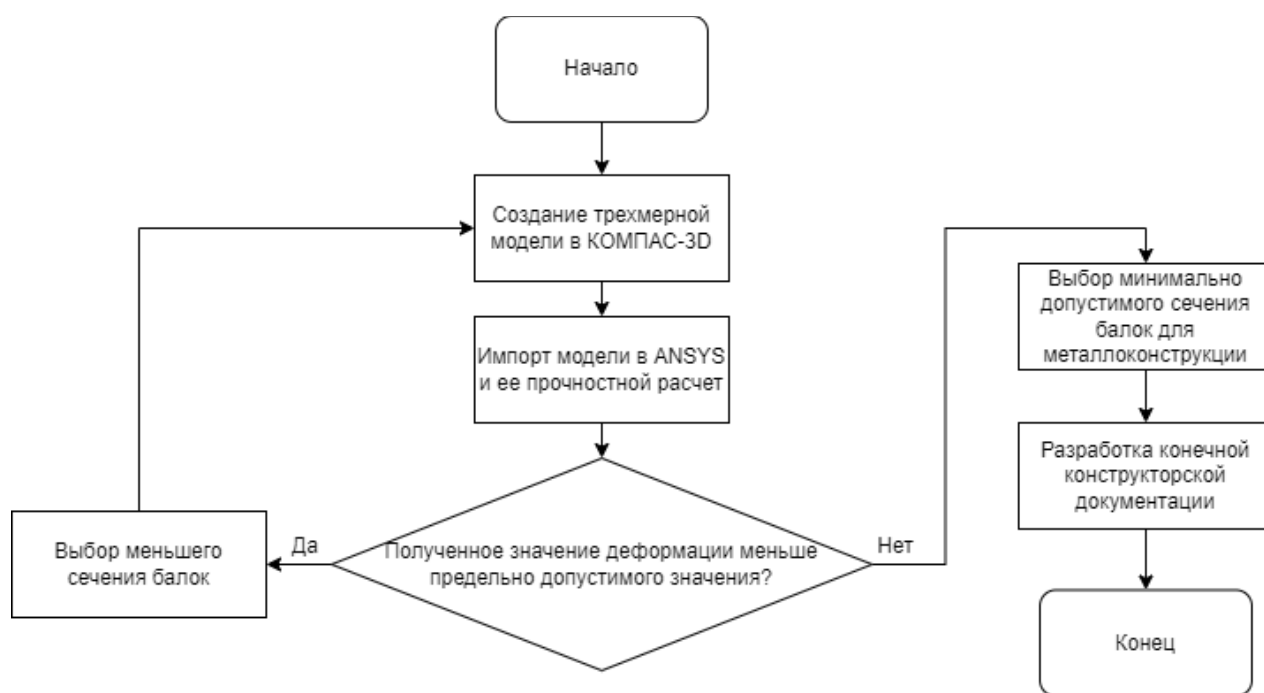


Рис. 1. Схема разработанной методики.

Разработанная методика предполагает совместную деятельность нескольких сотрудников предприятия в рамках одной программно-информационной системы. Диаграмма вариантов использования такой системы представлена на рисунке 2.

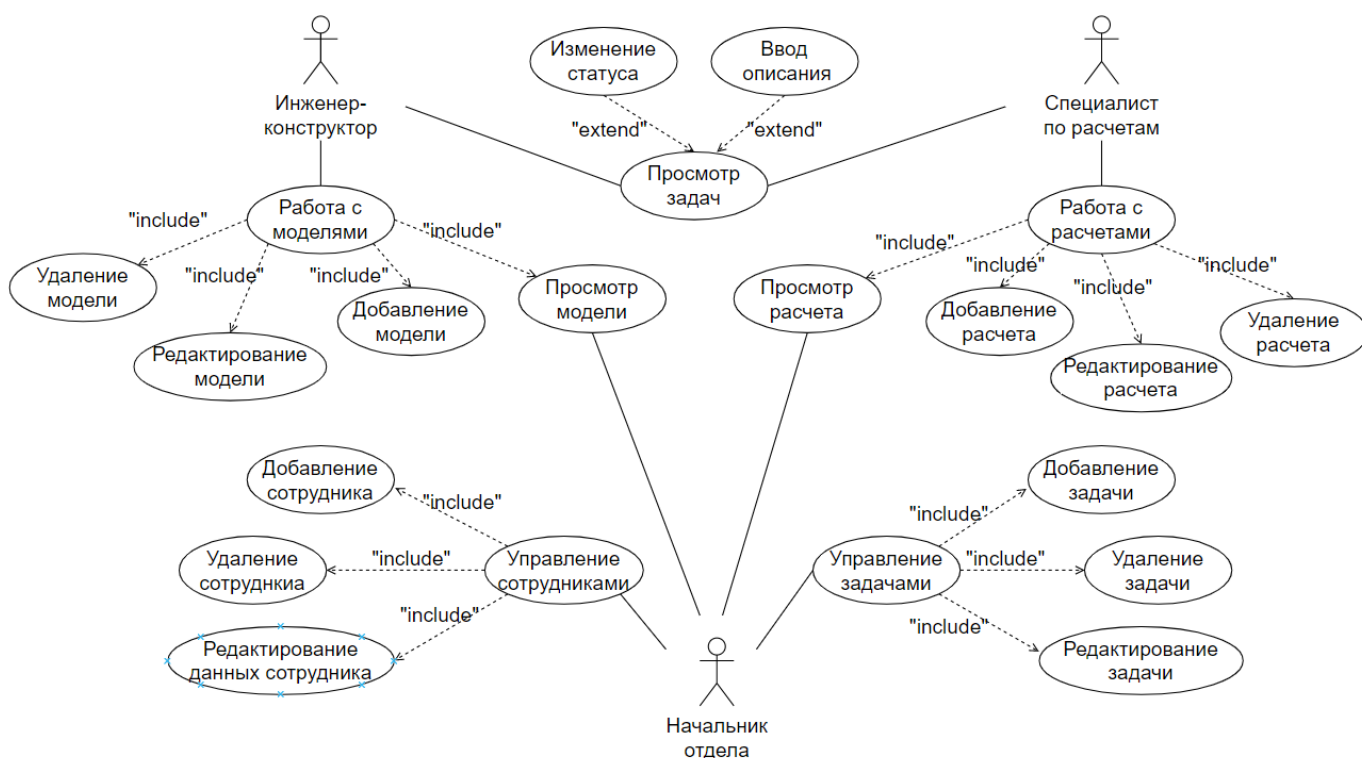


Рис. 2. Диаграмма вариантов использования разработанной системы.

Основными актерами являются:

- начальник отдела, который ставит задачи проектирования и принимает окончательное решение о выборе оптимальной модели для проектирования технологической оснастки;
- инженер-конструктор, выполняющий построение моделей в Компас-3D;
- специалист по расчетам, осуществляющий выполнение прочностных расчетов в ANSYS 2022.

Для инженерных расчетов будут использованы программные комплексы Компас-3D и ANSYS. Для реализации программно-информационной системы будем использовать Microsoft Visual Studio 2022 и Microsoft SQL Server.

Работа с программно-информационной системой начинается с постановки задач сотрудникам отдела. Начальник отдела заходит в свой профиль и в окне задачи заносит запись о новом проекте, при этом инженеру конструктору поставлена задача выполнить построение модели, а специалисту по расчетам – выполнить прочностной расчет. Описанные действия представлены на рисунках 3 и 4.

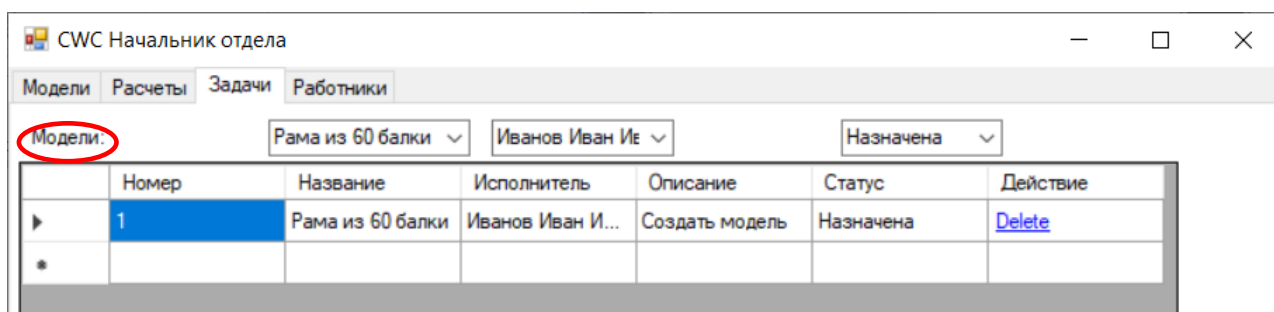


Рис. 3. Назначение задачи инженеру-конструктору.

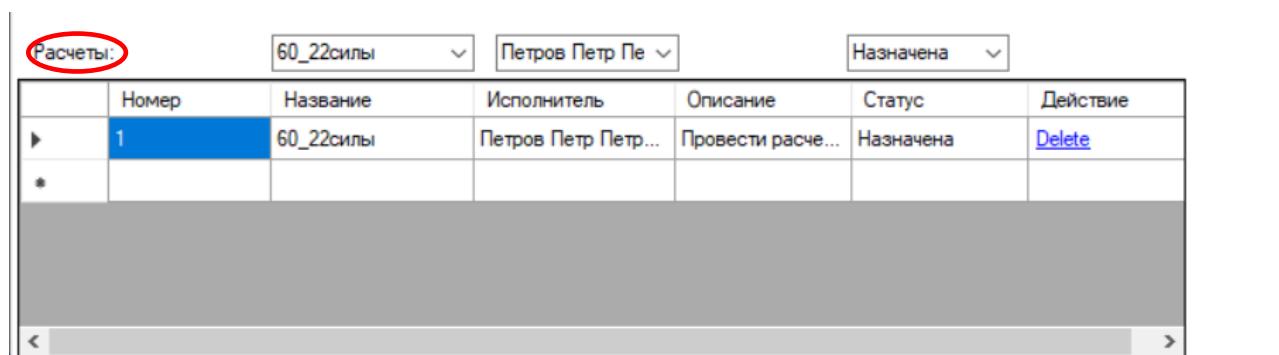


Рис. 4. Назначение задачи специалисту по расчетам.

Инженер конструктор авторизуется в программной системе, видит поставленную задачу, меняет статус задачи на принята и выполняет построение модели в системе Компас-3D. Эти действия представлены на рисунках 5 и 6.

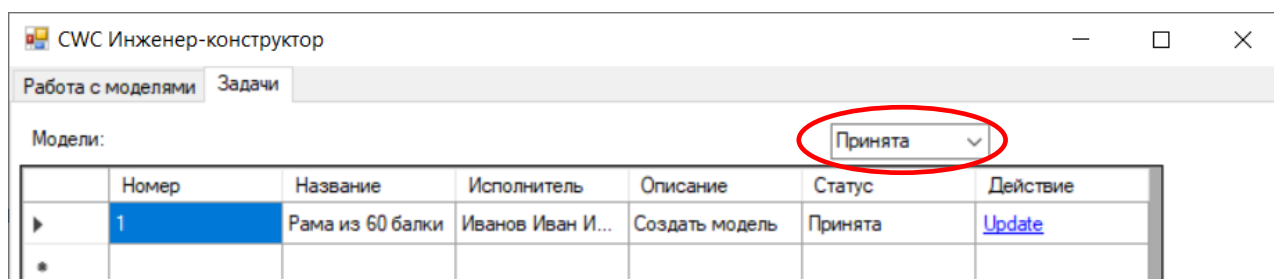


Рис. 5. Просмотр поставленных задач инженером-конструктором.

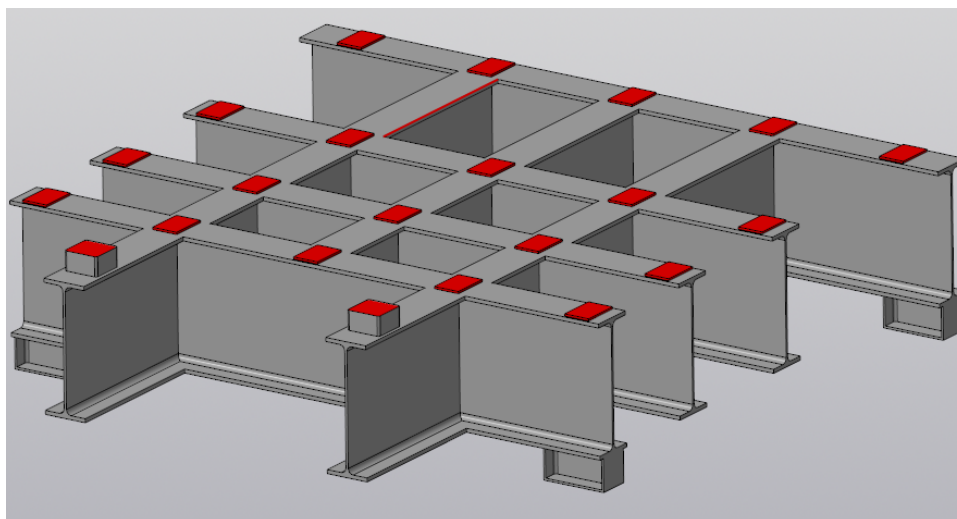


Рис. 6. Построение модели в Компас-3D.

Далее он переходит в окно «Работа с моделями» и нажимает кнопку «Открыть папку с моделями» для того чтобы перейти в ее расположение. В нее он загружает построенную модель, затем вносит информацию о спроектированной оснастке. После окончания работы над задачей инженер-конструктор меняет статус задачи на «Выполнена». Описанные действия представлены на рисунках 7 и 8.

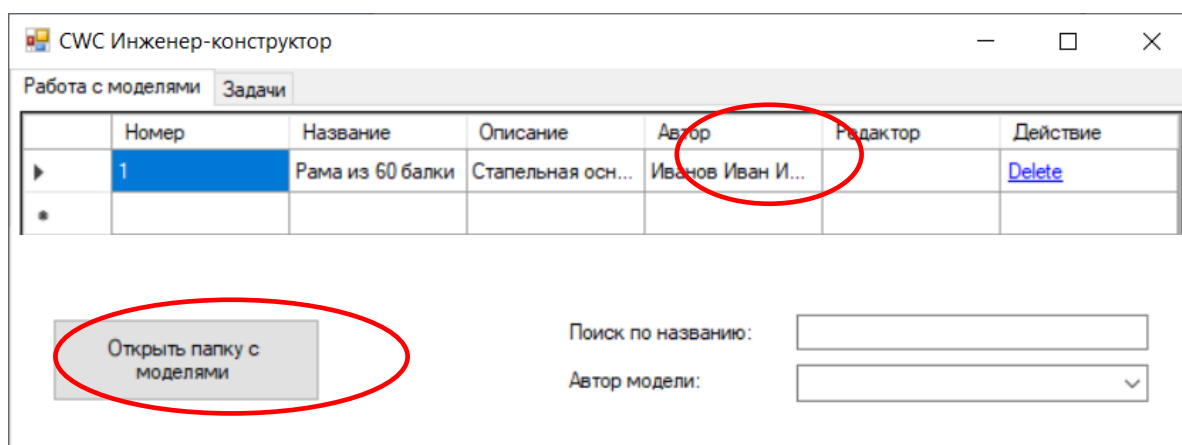


Рис. 7. Окно «Работа с моделями».

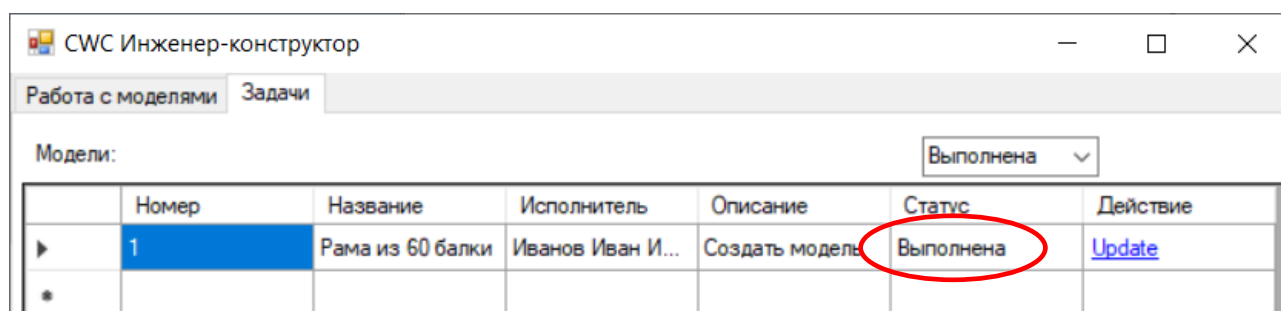


Рис. 8. Отметка о выполненной задаче.

Затем авторизацию в системе проходит специалист по расчетам. Он видит поставленную задачу, меняет статус задачи на принята, выполняет расчет модели в программном комплексе Ansys. Для этого модель из КОМПАС-3D будем импортировать в формате SAT в CAE-пакет ANSYS для дальнейшей работы с ней. Файл SAT - это 3D-модель, сохраненная в формате геометрического моделирования ACIS. Название формата расшифровывается как Standard ACIS Text – Стандартный текст ACIS. Файл содержит описание геометрии 3D-модели в текстовом формате ASCII и может быть просмотрен в простом текстовом редакторе [3].

Прочностные расчеты технологической оснастки, в частности стапеля сборки торцевой стены вагона-хоппера, спроектированной в Компас-3D, будем проводить в пакете программ ANSYS – передовом программном обеспечении для инженерного анализа и численного моделирования. После импортирования модели выполним построение сетки конечных элементов, а также зададим свойства, необходимые для расчета. Оснастка крепится к основанию с помощью четырех опор-ног. При расчете учитывается вес торцевой стены вагона-хоппера, для сборки которой и разрабатывается оснастка, силы, действующие со стороны 22-ух пневмоприжимов, с помощью которых фиксируются детали стены, а также гравитация.

Согласно конструкторской документации, максимально допустимое значение деформации стапельной оснастки при сборке торцевой стены вагона-хоппера составляет 2.5 мм. А как видно из результатов прочностного расчета, деформация составит всего 0.13 мм, что говорит об избыточной прочности проектируемого стапеля, которая ведет к необоснованному использованию материально-финансовых ресурсов. Схема расчета представлена на рисунке 9.

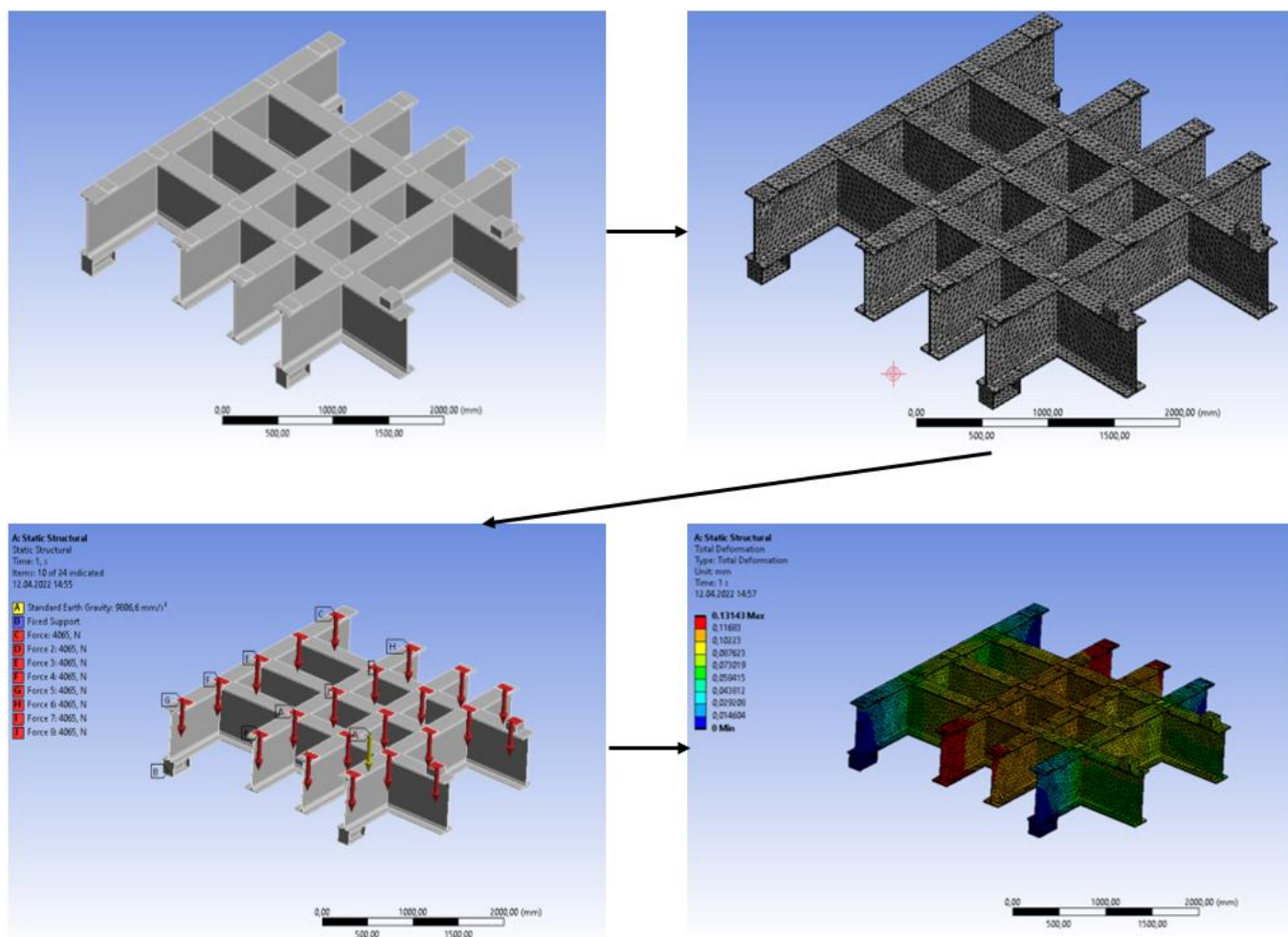


Рис. 9. Схема проводимого расчета.

Далее специалист по расчетам переходит в окно «Работа с расчетами» и нажимает кнопку «Открыть папку с расчетами для того чтобы перейти в ее расположение. В нее он загружает проект с выполненным расчетом, затем вносит информацию о нем. После окончания работы над задачей специалист по расчетам меняет статус задачи на «Выполнена». Эти действия представлены на рисунках 10 и 11.

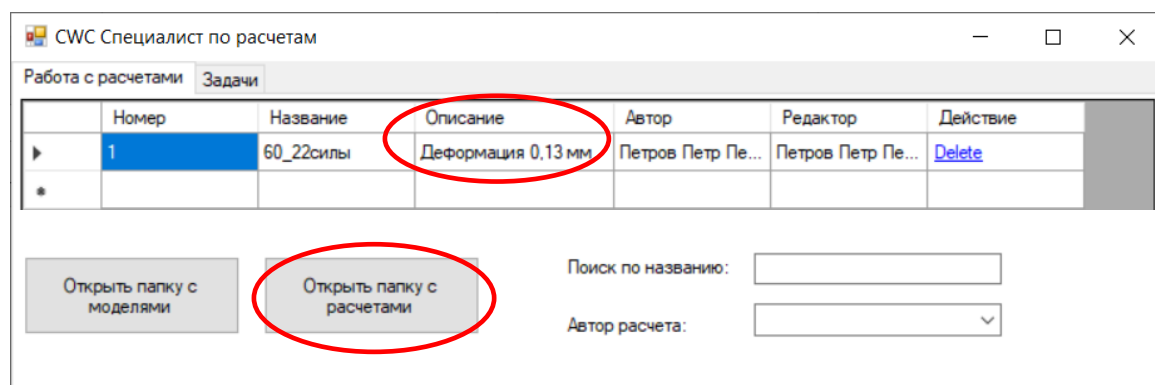


Рис. 10. Окно «Работа с расчетами».

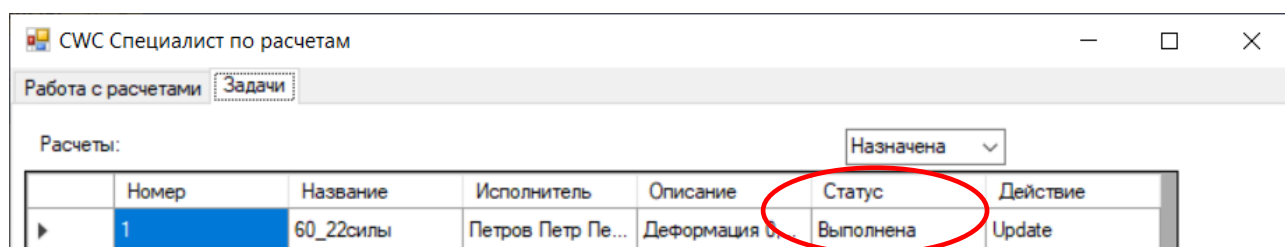


Рис. 11. Отметка специалиста по расчетам о выполненной задаче

Начальник отдела видит, что полученная деформация 0.13мм намного меньше предельно допустимого значения, равного 2.5мм и принимает решение о проведении расчетов с сечениями меньшего размера (рис. 12). Сечение балки представлено на рисунке 13. Всего будут рассмотрены 6 альтернативных моделей, размеры их балок приведены в таблице 1.

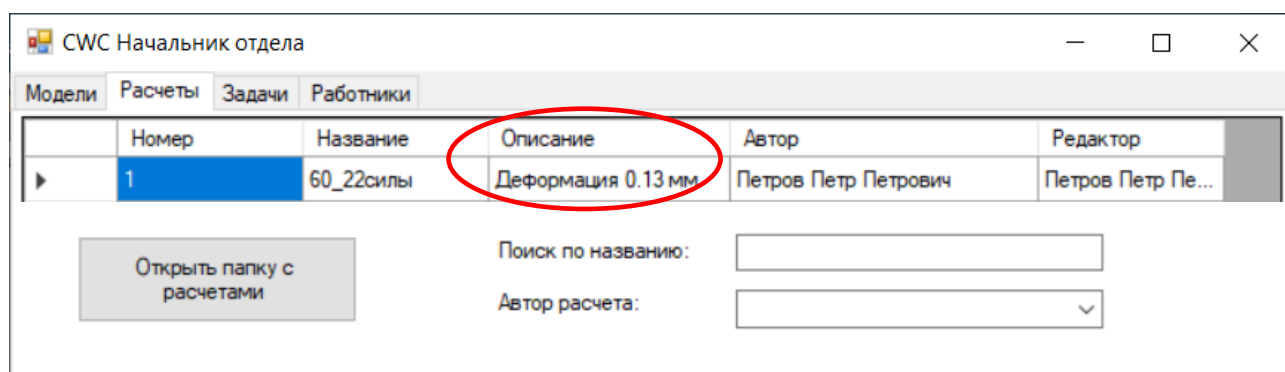


Рис. 12. Просмотр информации о моделях.

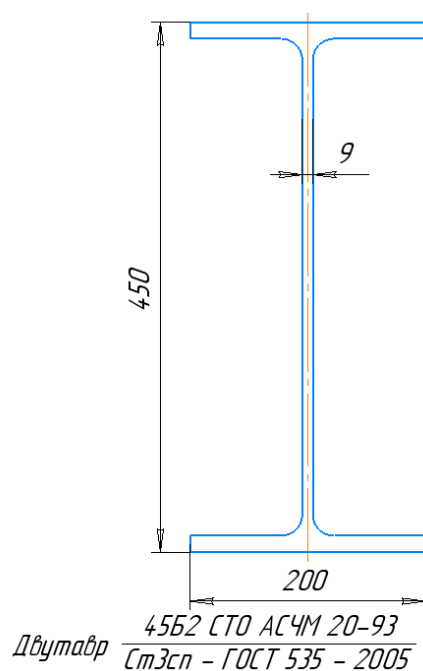


Рис. 13. Сечение балки.

Предлагаемые сечения для расчетов

Номер модели	Внешние балки	Внутренние балки
1	600*200	450*200
2	450*200	450*200
3	450*200	300*150
4	300*150	300*150
5	300*150	200*100
6	200*100	200*100

Начальник отдела назначает новые задачи по созданию и расчету моделей из балок, меньших сечений (рис. 14).

Скриншот интерфейса программы 'CWC Начальник отдела'. Вверху расположены вкладки: 'Модели', 'Расчеты', 'Задачи', 'Работники'. Вкладки 'Модели' и 'Расчеты' выделены красными кругами. Вкладка 'Модели' содержит следующие данные:

Номер	Название	Исполнитель	Описание	Статус	Действие
1	Рама из 60 балки	Иванов Иван И...	Создать модель	Выполнена	Delete
3	Рама из 60+45 ...	Федоров Андре...	Создать модель	Назначена	Delete
4	Рама из 45 балки	Сергеев Никола...	Создать модель	Назначена	Delete
5	Рама из 45+30 ...	Иванов Иван И...	Создать модель	Назначена	Delete
6	Рама из 30 балки	Федоров Андре...	Создать модель	Назначена	Delete

Вкладка 'Расчеты' содержит следующие данные:

Номер	Название	Исполнитель	Описание	Статус	Действие
1	60_22силы	Петров Петр Пе...	Деформация 0...	Выполнена	Delete
4	60+45_22силы	Андреев Макси...	Провести расчет	Назначена	Delete
5	45_22силы	Петров Петр Пе...	Провести расчет	Назначена	Delete
6	45+30_22силы	Андреев Макси...	Провести расчет	Назначена	Delete
7	30_22силы	Петров Петр Пе...	Провести расчет	Назначена	Delete

Рис. 14. Назначение новых задач.

После выполнения Начальник отдела видит всю информацию о моделях и расчетах, вносимую сотрудниками (рис. 15, 16).

CWC Начальник отдела

Номер	Название	Описание	Автор	Редактор
1	Рама из 60 балки	2192.7 кг	Иванов Иван Иванович	
10	Рама из 60+45 балок	1601.7 кг	Федоров Андрей Николаевич	
11	Рама из 45 балки	1340.4 кг	Сергеев Николай Петрович	
12	Рама из 45+30 балок	1231.6 кг	Иванов Иван Иванович	
13	Рама из 30 балки	837.6 кг	Федоров Андрей Николаевич	
14	Рама из 30+20 балок	702.0 кг	Сергеев Николай Петрович	
15	Рама из 20 балки	512.3 кг	Иванов Иван Иванович	

Открыть папку с моделями

Поиск по названию:

Автор модели:

Рис. 15. Информация о моделях (вес полученных конструкций).

CWC Начальник отдела

Номер	Название	Описание	Автор	Редактор
1	60_22силы	Деформация 0.13 мм	Петров Петр Петрович	Петров Петр Пе...
10	60+45_22силы	Деформация 0.20 мм	Андреев Максим Витальевич	
11	45_22силы	Деформация 0.28 мм	Петров Петр Петрович	
12	45+30_22силы	Деформация 0.33 мм	Андреев Максим Витальевич	
13	30_22силы	Деформация 0.76 мм	Петров Петр Петрович	
14	30+20_22силы	Деформация 1.04 мм	Андреев Максим Витальевич	
15	20_22силы	Деформация 2.52 мм	Петров Петр Петрович	

Открыть папку с расчетами

Поиск по названию:

Автор расчета:

Рис. 16. Информация о расчетах (деформация в местах фиксации собираемого узла вагона).

Полученные значения деформаций и масс альтернативных моделей занесем в таблицу 2.

Сравнение характеристик разработанных моделей

Сечение	60	60+45	45	45+30	30	30+20	20
Значение Total Deformation, мм	0,13	0,20	0,28	0,33	0,76	1,04	2,53
Масса модели, кг	2192,7	1601,7	1340,4	1231,6	837,6	702,0	518,2

В результате выберем оптимальный вариант – комбинированную модель из 30-ой и 20-ой балок, деформация которой составит 1.04 мм. Сравним модели по массе и стоимости используемых материалов, чтобы посчитать экономию. Масса начальной модели составляет 2192.7 кг, а полученной оптимальной – 702.0 кг. Стоимость двутавровых балок для начальной модели ступельной оснастки составит 206 497 рублей, а для полученной оптимальной 64 084 рубля. Экономия составит 142 413 рублей.

Таким образом, разработанная методика автоматизированного расчета технологической оснастки показала свою эффективность.

СПИСОК ЛИТЕРАТУРЫ

1. Что такое технологическая оснастка [Электронный ресурс]. – Режим доступа: <http://www.miriso.ru/article/20170014.html> (дата обращения: 12.10.2022).
2. Проектирование и расчет технологической оснастки [Электронный ресурс]. – Режим доступа: <https://cyberpedia.su/14x1015e.html?> (дата обращения: 17.10.2022).
3. Файл SAT – чем открыть, описание формата [Электронный ресурс]. – Режим доступа: <https://open-file.ru/types/sat> (дата обращения: 25.10.2022).

ПИМАСHEВ В. А., НЕЛИН О. А., ФИРСОВА С. А.
РАЗРАБОТКА ПРОГРАММНО-ИНФОРМАЦИОННОЙ СИСТЕМЫ
ДЛЯ АВТОМАТИЗИРОВАННОГО УЧЕТА
ТРЕНИРОВОЧНОГО ПРОЦЕССА СПОРТСМЕНОВ-БИАТЛОНИСТОВ
НА ОСНОВЕ ИХ ЭЛЕКТРОННЫХ ДНЕВНИКОВ

Аннотация. В статье приводится описание программно-информационной системы (ПИС), предназначенной для автоматизированного учета показателей тренировочного процесса спортсменов на основе данных, получаемых со специальных пульсометров. Данные передаются в электронный дневник спортсмена, созданный с помощью сервиса Google Sheets, а затем обрабатываются в ПИС для учета и анализа выполненной спортсменами работы в течение годового цикла.

Ключевые слова: электронный дневник спортсмена, Google Sheets, тренировочный процесс, биатлон, пульсометр.

PIMASHEV V. A., NELIN O. A., FIRSOVA S. A.
DEVELOPMENT OF SOFTWARE AND INFORMATION SYSTEM
FOR AUTOMATED ACCOUNTING OF BIATHLON ATHLETES TRAINING
USING DATA OF THEIR ELECTRONIC DIARIES

Abstract. The article describes a software and information system designed for automated accounting of athletes' training process indicators based on data obtained from special heart rate monitors. The data is transmitted to the athlete's electronic diary, created using the Google Sheets service, and then processed in the software and information system to record and analyze the work performed by athletes during the annual cycle.

Keywords: athlete's electronic diary, Google Sheets, training process, biathlon, heart rate monitor.

Дневник спортсмена предназначен для спортсменов и тренеров в целях ведения индивидуального контроля над тренировочным процессом и наблюдения за динамикой широкого круга показателей двигательной деятельности занимающегося в течение годичного тренировочного цикла.

Спортивный дневник необходим детям, планирующим долгосрочную спортивную карьеру, он помогает отслеживать физическое состояние и спортивные достижения, а также является лучшей формой самоконтроля спортсмена. В дневнике рекомендуется фиксировать объем и интенсивность ежедневных физических нагрузок, результаты контрольных стартов и соревнований, некоторые объективные и субъективные показатели состояния организма в

период тренировочного процесса. С помощью дневника можно отследить, когда появились первые признаки спада, разобраться, с чем они могли быть связаны. Также можно сравнить результаты текущего и прошлых сезонов, свои показатели с показателями других спортсменов. Поэтому ведение спортивного дневника является очень важной составляющей прогресса в спорте.

В настоящее время в ГАУ РМ «СШОР по зимним видам спорта» разработан электронный дневник спортсмена, который находится на стадии тестирования и внедрения в команде старших девушек сборной Республики Мордовия по биатлону. Дневник разработан на основе бесплатного сервиса для работы с электронными таблицами Google Sheets, или Google-таблицы. Основными преимуществами применения Google-таблиц, используемыми при разработке электронного дневника, являются [1]:

- сохранение огромных массивов данных;
- бесплатная работа с сервисом через браузер и мобильные приложения;
- совместная работа над документами, предусматривающая настройку прав доступа на редактирование, комментирование или чтение для человека, которому открыт доступ к таблице по ссылке;
- защита от редактирования определенных диапазонов данных;
- большое количество встроенных функций для обработки документов с возможностью создания собственных функций;
- возможность сохранения истории изменений документа и отката в любую точку, где необходимо внести новые данные, изменить формулу расчетов, добавить столбец и т. п.;
- хранение случайно удаленных документов в «корзине» в течение 30 дней;
- импорт данных из других доступных источников без потери функционала.

Каждый лист Google-таблицы доступен для редактирования конкретным спортсменом. Лист имеет следующую структуру:

- 1) столбцы таблицы содержат описание разделов спортивной подготовки: циклическая работа, которая включает ходьбу и ходьбу с палками, бег и бег с палками, прыжковую имитацию лыжных ходов с палками, тренировка с использованием лыж, лыжероллеров, велосипеда; спортивные игры; силовая подготовка, которая включает силовую работу на лыжах, лыжероллерах и в тренажерном зале; строки таблицы содержат информацию о количестве пройденных километров и затраченном времени по каждому разделу на зарядке, утренней и вечерней тренировках; в конце каждого тренировочного дня ведется подведение итогов проделанной спортсменом работы (рис. 1, 2);

Электронный дневник спортсмена ☆ 📁 ☁ Файл Правка Вид Вставка Формат Данные Инструменты Расширения Справка Последнее изменение ↶ ↷ 🖨 🔍 100% р. % .0 .00 123 Calibri 11 В I U A 🔗 🔧 🔢 ☰ K1324 fx

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1302															
1303		Циклическая работа													
1304	Дата	Ходьба, ходьба с палками		Бег		Бег с палками		Имитация		Лыжероллеры		Лыжи		Велосипед	
1305	06.09.2022	мин	км	мин	км	мин	км	мин	км	мин	км	мин	км	мин	км
1306	Зарядка														
1307	Утро 1 тр.									43	5,23				
1308	Вечер 2 тр.									17,3	4,5				
1309	Итого	0	0	0	0	0	0	0	0	60,3	9,73	0	0	0	0
1310															
1311		Циклическая работа													
1312	Дата	Ходьба, ходьба с палками		Бег		Бег с палками		Имитация		Лыжероллеры		Лыжи		Велосипед	
1313	07.09.2022	мин	км	мин	км	мин	км	мин	км	мин	км	мин	км	мин	км
1314	Зарядка														
1315	Утро 1 тр.			72	9,89										
1316	Вечер 2 тр.									50	8				
1317	Итого	0	0	72	9,89	0	0	0	0	50	8	0	0	0	0

Рис. 1. Циклическая работа в электронном дневнике.

	A	Q	R	S	T	U
1	Силовая подготовка					
2	Дата	Силовая работа на лыжах		Силовая работа на роллерах		Силовая работа в зале
1354	12.09.2022	мин	км	мин	км	мин
1355	Зарядка					10
1356	Утро 1 тр.					
1357	Вечер 2 тр.			55	8,46	
1358	Итого	0	0	55	8,46	10

Рис. 2. Силовая подготовка в электронном дневнике.

2) следующие столбцы содержат информацию о стрелковой подготовке: время холостого тренажа; общее количество выстрелов, количество попаданий и промахов, а также процент попаданий из положений лежа и стоя (рис. 3);

	A	V	W	X	Y	Z	AA	AB	AC
1303	Стрелковая подготовка								
1304	Дата	Холостой тренаж	Количество выстрелов	Стоя	Лежа	Промехи стоя	Промехи лежа	% попаданий стоя	% попаданий лежа
1305	06.09.2022	мин	кол-во	кол-во	кол-во	кол-во	кол-во	%	%
1306	Зарядка							0%	0%
1307	Утро 1 тр.		40	5	5	3	3	40%	40%
1308	Вечер 2 тр.							0%	0%
1309	Итого	0	40	5	5	3	3	40%	40%
1310									
1311	Стрелковая подготовка								
1312	Дата	Холостой тренаж	Количество выстрелов	Стоя	Лежа	Промехи стоя	Промехи лежа	% попаданий стоя	% попаданий лежа
1313	07.09.2022	мин	кол-во	кол-во	кол-во	кол-во	кол-во	%	%
1314	Зарядка							0%	0%
1315	Утро 1 тр.	50	50	20	20	12	2	40%	90%
1316	Вечер 2 тр.							0%	0%
1317	Итого	50	50	20	20	12	2	40%	90%

Рис. 3. Стрелковая подготовка в электронном дневнике.

3) последние столбцы содержат информацию о самочувствии спортсмена, которое оценивает сам спортсмен по 10-ти балльной шкале; количество минут, проведенных во время тренировки в разных зонах интенсивности (рис. 4).

1311		Самочувствие		1 зона	2 зона	3 зона	4 зона	5 зона	Итого
1312	Дата	1-10 баллов		мин	мин	мин	мин	мин	
1313	07.09.2022		Зарядка						0
1314	Зарядка		Утро 1 тр.	5	18	30	14		67
1315	Утро 1 тр.	2	Вечер 2 тр.	29	12	9			50
1316	Вечер 2 тр.	1	Итого	34	30	39	14	0	117
1317	Итого		%	29%	26%	33%	12%	0%	100%

Рис. 4. Зоны интенсивности и самочувствие в электронном дневнике.

Для получения информации, необходимой при заполнении электронного дневника, каждый спортсмен должен иметь специальный пульсометр со встроенным GPS-навигатором и поддержкой соединения со смартфоном, например, Polar M400 или Polar V800 [2], и нагрудный ремень со встроенным высокоточным датчиком частоты сердечных сокращений, например, Polar H7 (рис. 5) [3].



Рис. 5. Пульсометр Polar V400 с ремнем.

Включение пульсометра производится в момент начала тренировки, а отключение – после ее завершения. При этом после синхронизации датчика и пульсометра в приложении Polar Flow [4] появляется вся информация о ходе тренировочного процесса. Например, на рисунке 6 видно, что 07.09.2022 г. у спортсмена было две тренировки – утренняя и вечерняя.

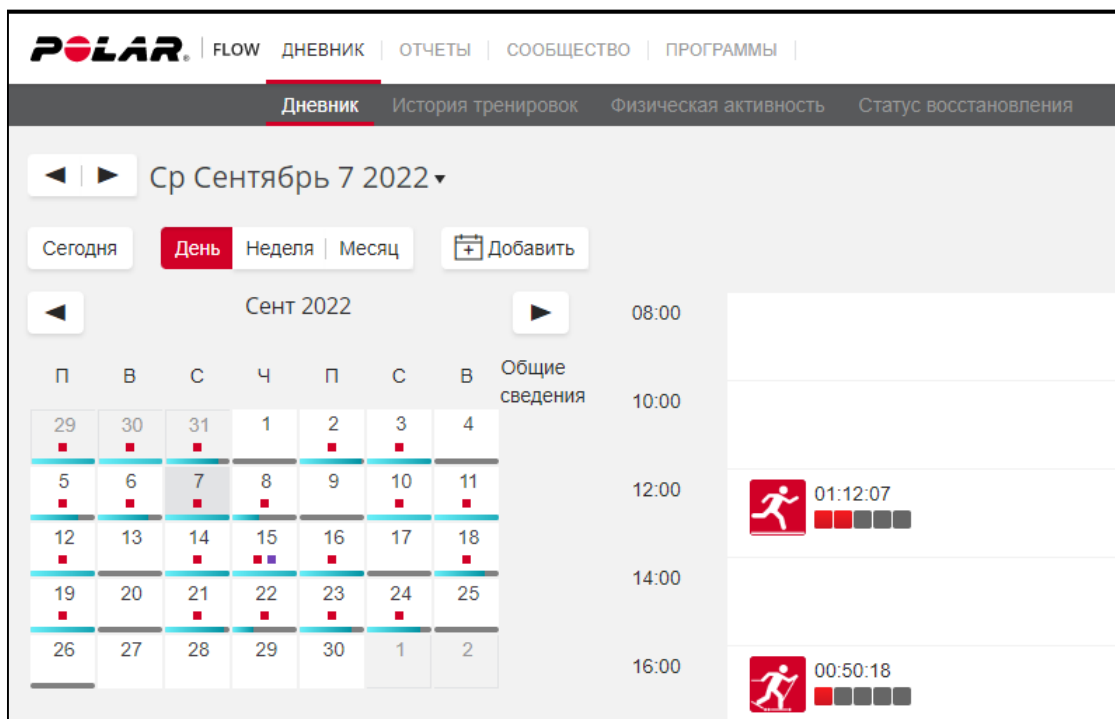


Рис. 6. Дневник в Polar Flow.

При выборе одной из тренировок можно увидеть её продолжительность и пройденную дистанцию, эти значения заносятся в соответствующие разделы электронного дневника (см. рис. 7).

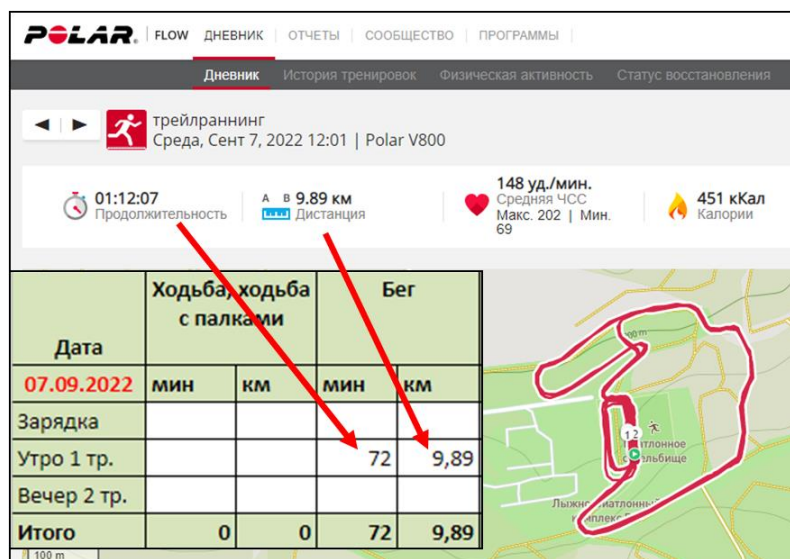


Рис. 7. Продолжительность и дистанция в Polar Flow.

Также можно увидеть количество минут, проведенных во время тренировки в разных зонах интенсивности (см. рис. 8).

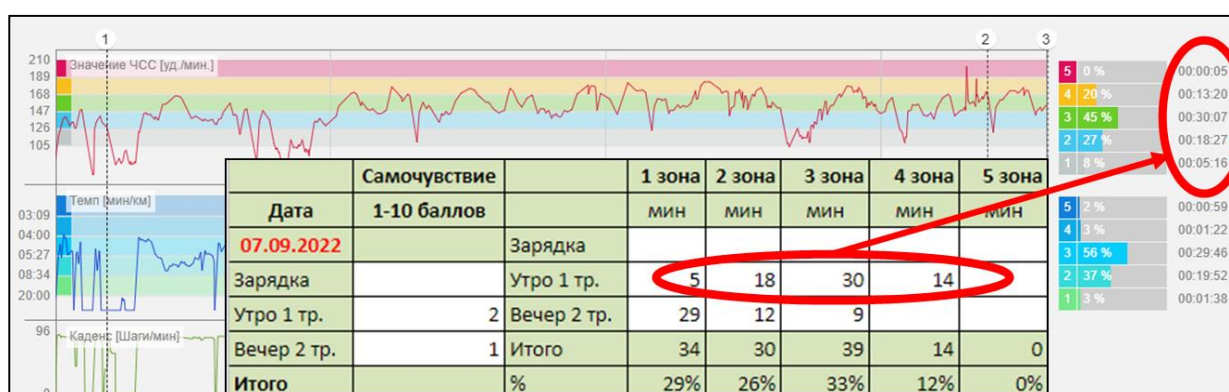


Рис. 8. Зоны интенсивности в Polar Flow.

В конце каждого месяца подводятся итоги по каждому виду работы. Например, итоговые значения о работе в различных зонах интенсивности за месяц представлены на рисунке 9.

	A	AI	AJ	AK	AL	AM	AN
1		1 зона	2 зона	3 зона	4 зона	5 зона	Итого
2	Дата	мин	мин	мин	мин	мин	
1259							
1260	Итого за месяц, в мин	320,00	1114,00	730,00	129,00	9,00	2302,00
1261	Итого за месяц, в часах	5,3	18,6	12,2	2,2	0,2	38,4
1262		13,90%	48,39%	31,71%	5,60%	0,39%	100,00%

Рис. 9. Подведение итогов о работе в различных зонах интенсивности за месяц.

Все данные, представленные в электронном дневнике, передаются в программно-информационную систему для их последующей автоматизированной обработки и анализа.

Для реализации программно-информационной системы были использованы следующие инструментальные средства:

- 1) редактор кода – Visual Studio Code (1.72);
- 2) язык программирования Python (3.10.2) совместно с библиотеками:
 - NumPy – библиотека с открытым исходным кодом, поддерживающая высокоуровневые математические функции, предназначенные для работы с многомерными массивами;
 - PyQt6 – библиотека для создания приложений с графическим интерфейсом с помощью инструментария Qt;
 - Matplotlib – библиотека для визуализации данных двухмерной и трехмерной графикой;
 - datetime – модуль, предоставляющий классы для обработки времени и даты;
 - httplib2 – комплексная клиентская библиотека HTTP;
 - oauth2client.service_account – библиотека, позволяющая работать с API Google;
- 3) текстовый формат обмена данными для авторизации в Google API – формат JSON.

При запуске программно-информационной системы открывается окно приложения, представленное на рисунке 10.

**УЧЕТ ТРЕНИРОВОЧНОГО ПРОЦЕССА
СПОРТСМЕНОВ-БИАТЛОНISTОВ
НА ОСНОВЕ ИХ ЭЛЕКТРОННЫХ
ДНЕВНИКОВ**

**Для начала работы получите
данные с сервера**

**Задайте критерии для
получения отчета:**

Стрелковая подготовка:

Зоны интенсивности:

Получить данные с сервера

ФИО спортсмена:

Временной промежуток:

Тип тренировки:

Построить график

Получить отчет

Рис. 10. Окно приложения.

Для начала работы пользователю необходимо получить данные с сервера. Код этой функции представлен на рисунке 11.

```
def clicked():
    global users
    global length
    global rows
    global arr
    for user in users:
        values = service.spreadsheets().values().get
        (spreadsheetId=sheet_id,range=user,majorDimension='ROWS').execute()
        rows.append(values.get('values'))
    length=len(rows)
    arr=parse(rows)
```

Рис. 11. Код функции, отвечающей за получение данных с сервера.

В данной функции для каждого спортсмена, зарегистрированного в программно-информационной системе, происходит поиск совпадения его фамилии с именем листа Google-таблицы. Для выполнения соответствующего запроса необходимо указать параметр `spreadsheetId`, содержащий уникальный идентификатор таблицы, который можно получить в адресной строке браузера. Далее запрос отправляется на сервер; полученные от сервера строки Google-таблицы сохраняются в массиве, обрабатываются для более удобного использования методом `parse()`, а затем выводятся в консоль. Например, для одного дня имеем следующую обработанную запись:

```
[datetime.date(22,2,18),'0','0','17','1,63','0','0','0','0','0','0','1
34','22,89','0','0','0','0','0','0','0','0','30','0','0','0','2','
8','12','13','114']
```

Рис. 12. Вывод в консоль данных, обработанных приложением.

Программно-информационная система имеет широкий спектр возможностей анализа данных, сгруппированных по различным критериям: временным интервалам (месяц, квартал, год, сезон), возрастным группам (группы «Мужчины и женщины», «Юниоры и юниорки», «Старшие юноши и девушки» и т. д.), видам спортивной подготовки (циклическая и силовая работа, стрелковая подготовка и т. д.). Также подробный анализ можно провести как для одного спортсмена, так и для группы с целью сравнения нужных показателей. Получаемые результаты для большей наглядности визуализируются с помощью построения диаграмм.

Рассмотрим некоторые из описанных возможностей.

На рисунке 13 представлена гистограмма об объеме циклической работы, выполненной спортсменом за месяц.

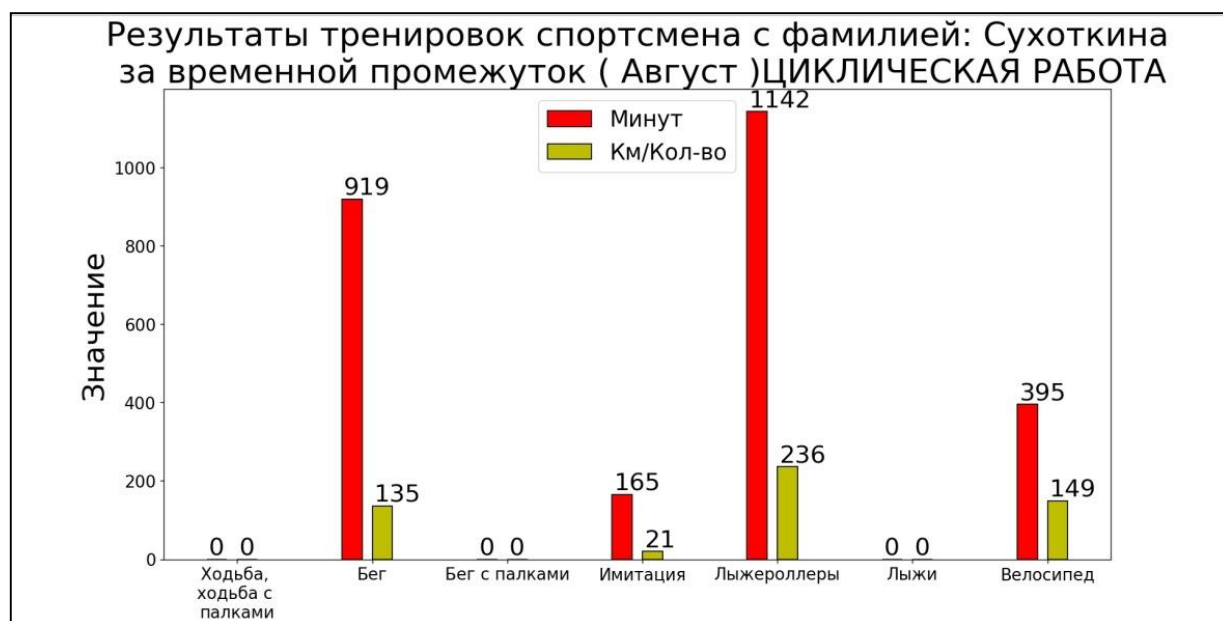


Рис. 13. Гистограмма об объемах циклической работы спортсмена за месяц.

Из рисунка 13 видно, что в августе преобладали тренировки с использованием лыжероллеров, общий объем этой работы составил 1142 минуты или 19 часов, при этом было пройдено 236 км.

Можно проводить сравнительный анализ любой проделанной спортсменом работы по выбранным месяцам, например, для показателя точности стрельбы (рис. 14).

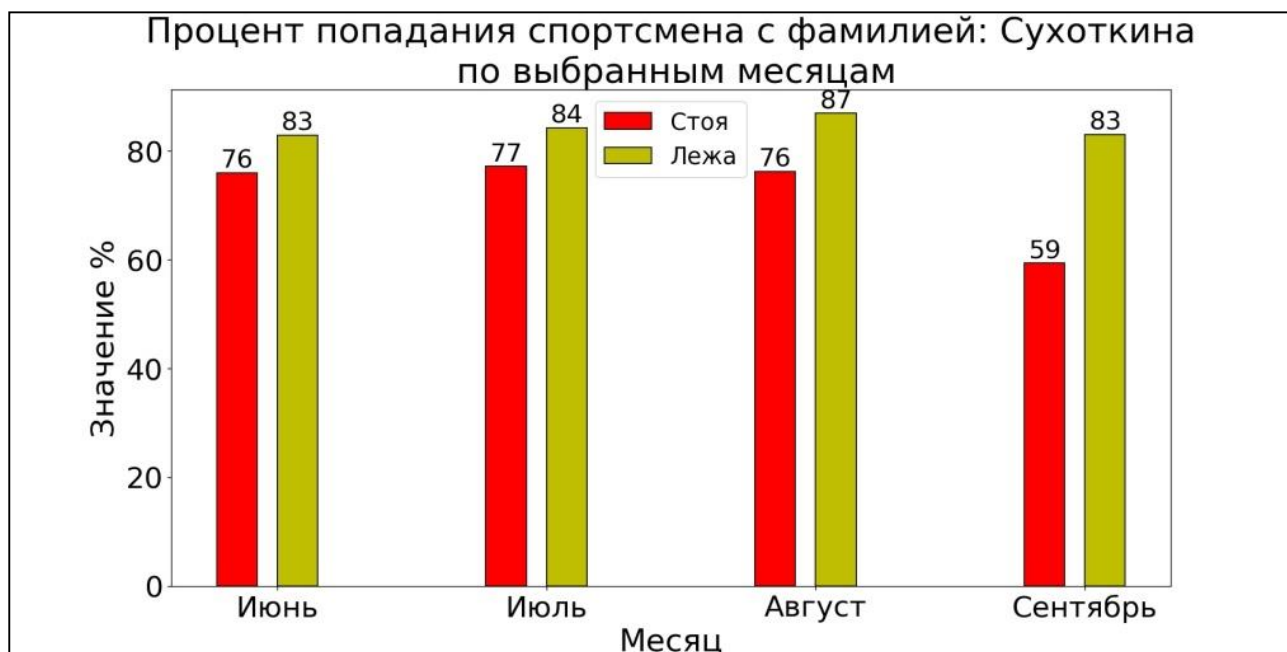


Рис. 14. Точность стрельбы спортсмена из различных положений по месяцам.

Из рисунка 14 можно сделать следующий вывод: спортсмен показывал стабильные результаты стрельбы из положения стоя на протяжении трех летних месяцев, однако в сентябре процент попадания упал приблизительно на 20%, что может быть объяснено неустойчивым психо-эмоциональным состоянием спортсмена, связанным с подготовкой к главному старту летнего сезона – Первенству России по летнему биатлону и началом нового учебного года в школе.

На рисунке 15 приведена гистограмма об объемах силовой работы спортсмена, выполненной в подготовительном периоде (июль) и предсоревновательном периоде (август). Видно, в июле силовая работа в зале преобладала над остальными видами силовой нагрузки и носила развивающий характер, что свойственно для подготовительного периода. В августе она была снижена в 2 раза и носила тонизирующий характер, в тоже время объем силовой работы на лыжероллерах увеличился в 4 раза с целью адаптации мышц к движениям на лыжероллерах.

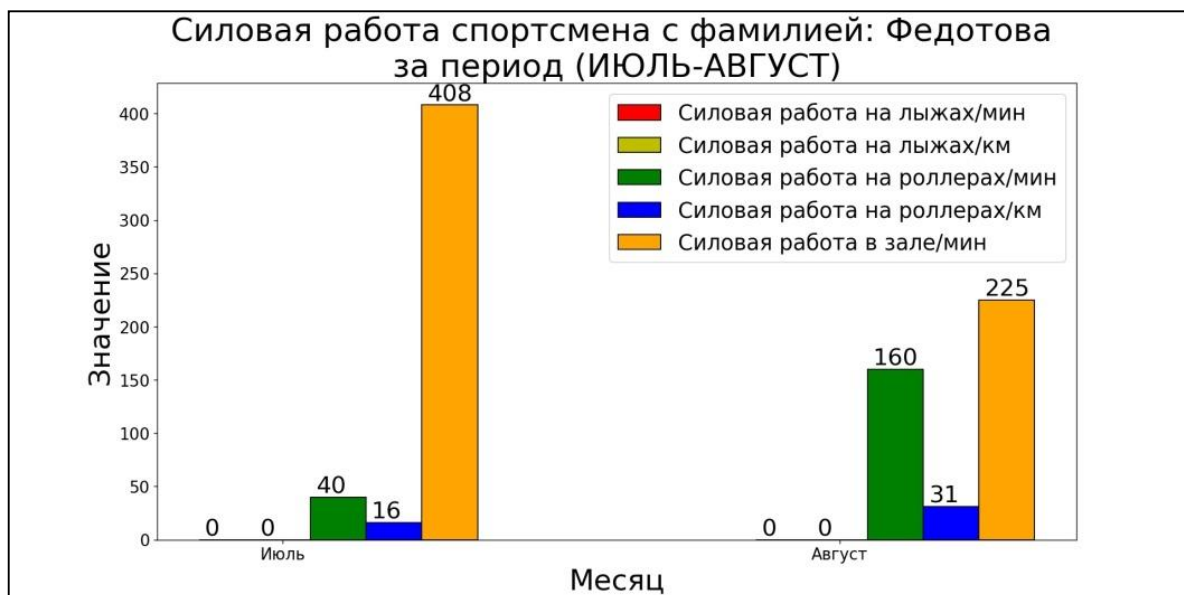


Рис. 15. Объемы силовой работы спортсмена по месяцам.

В программной системе присутствует возможность построения графиков для различных показателей по всем спортсменам с целью сравнения выполненных ими объемов работ относительно друг друга (рис. 16).

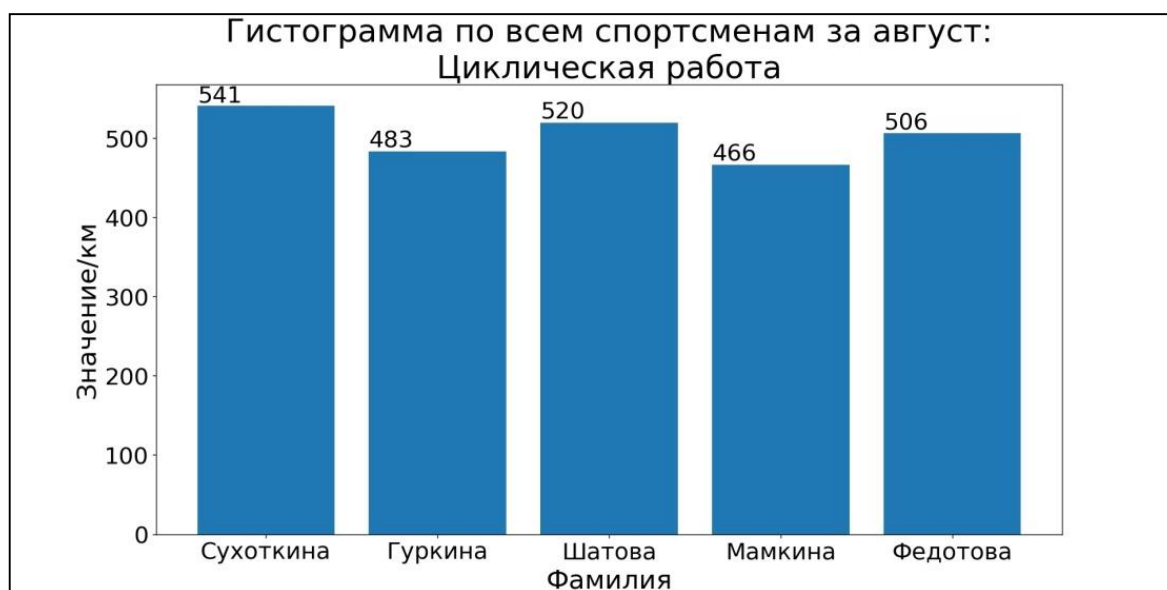


Рис. 16. Объемы циклической работы спортсменов за месяц.

На рисунке 16 приведена гистограмма, которая показывает, что в августе спортсмены имели различные относительные показатели по зонам интенсивности во время выполнения тренировочного процесса.

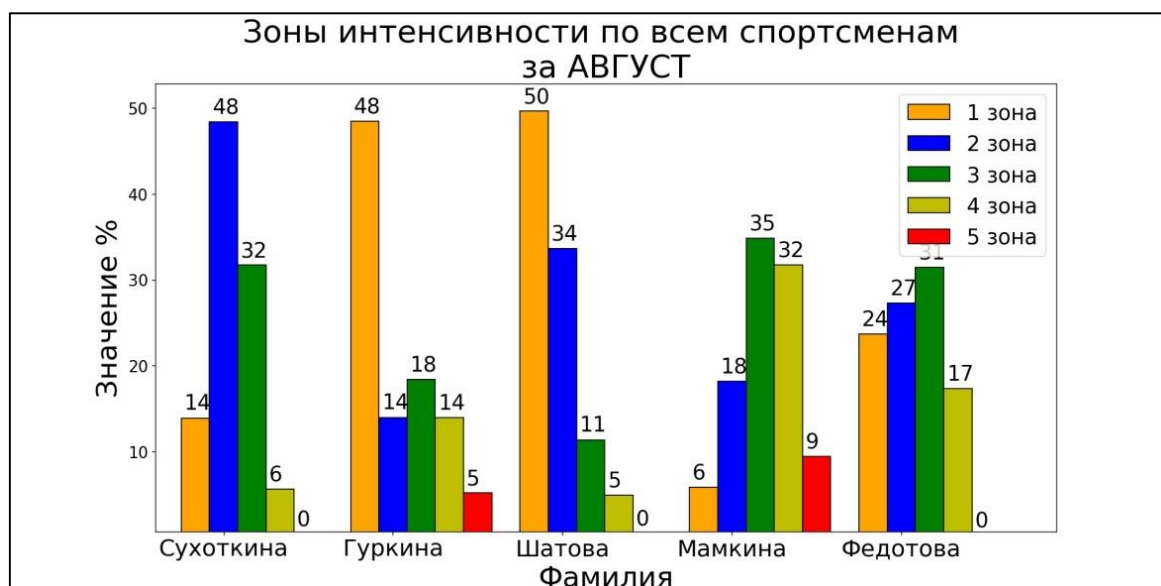


Рис. 17. Зоны интенсивности у спортсменов.

Гистограмма на рисунке 17 показывает, что у спортсменок Гуркиной и Шатовой, которые выступают в возрастной группе «Юниорки», в значительной степени преобладает 1-ая зона интенсивности, что говорит о более сильных изменениях в сердце (размеры сердца увеличиваются) и более экономичном расходе энергии. При этом частота сокращений сердца не сильно повышена, а сокращающая способность миокарда сильнее. Другие спортсменки Мамкина и Федотова, выступающие в возрастной группе «Младшие девушки», имеют менее развитое сердце, поэтому их показатели ЧСС являются более высокими и находятся преимущественно в 3-ей зоне интенсивности. Однако, у их ровесницы Сухоткиной, преобладает 2-ая зона интенсивности, что говорит о уже заметных изменениях в сердце. Эти изменения также влияют и на спортивный результат, о чём свидетельствуют отличные результаты этой спортсменки на Российских соревнованиях в своём возрасте.

Таким образом, в статье приводится описание разработанной авторами программно-информационной системы, предназначенной для автоматизированного учета показателей тренировочного процесса спортсменов-биатлонистов на основе их электронных дневников, созданных с использованием бесплатного сервиса Google Sheets. Необходимые данные для ведения дневника каждый спортсмен получает, исходя из информации от специальных пульсометров со встроенным GPS-навигатором. Программно-информационная система имеет широкий спектр возможностей для анализа данных, сгруппированных по различным критериям. Получаемые результаты для большей наглядности визуализируются с помощью построения диаграмм. В статье также сформулированы некоторые из выводов, которые можно получить на основе проведенного анализа. Выводы можно использовать для корректировки планов тренировочного процесса на следующий соревновательный период.

СПИСОК ЛИТЕРАТУРЫ

1. Обзор возможностей Google Sheets [Электронный ресурс]. – Режим доступа: <https://www.google.ru/intl/ru/sheets/about/> (дата обращения: 12.10.2022).
2. Пульсометр Polar M400 – подробный обзор [Электронный ресурс]. – Режим доступа: <https://training365.ru/pulsometr-polar-m400-podrobnyj-obzor/> (дата обращения: 14.10.2022).
3. Обзор: монитор сердечного ритма – пульсометр Polar H7 [Электронный ресурс]. – Режим доступа: <https://medgadgets.ru/fitness/obzor-monitor-serdechnogo-ritma-pulsometr-polar-h7.html> (дата обращения: 16.10.2022).
4. Polar Flow [Электронный ресурс]. – Режим доступа: <https://flow.polar.com/> (дата обращения: 20.10.2022).

ЕГОРОВА Д. К., ГАРИН М. А.

РЕАЛИЗАЦИЯ ЗАДАЧИ КЛАСТЕРИЗАЦИИ ВОЗДУШНЫХ СУДОВ В KNIME ANALYTICS PLATFORM

Аннотация. В статье приведено создание рабочего процесса аналитической платформы KNIME Analytics Platform. Выполнена кластеризация и визуализация данных в режиме реального времени.

Ключевые слова: KNIME Analytics Platform, кластеризация, визуализация, воздушные суда.

EGOROVA D. K., GARIN M. A.

IMPLEMENTATION OF AIRCRAFT CLUSTERING PROBLEM ON KNIME ANALYTICS PLATFORM

Abstract. The creation of the KNIME Analytics Platform workflow is given in the article. Clustering and visualization of data in real time is performed.

Keywords: KNIME Analytics Platform, clustering, visualization, aircraft.

В работе приведена реализация алгоритма четкой кластеризации воздушных судов по заданным параметрам в режиме реального времени из источников с открытым API в KNIME Analytics Platform [3].

Технология ADS-B.

Будем использовать большие данные полученные с ресурса с открытым API Flightradar24. Flightradar24 – это система отслеживающая авиасообщение всего мира в режиме реального времени. Здесь аккумулированы данные технологий ADS-B, MLAT, авиа-радаров и сведения о расписании, статусе рейсов от авиакомпаний и аэропортов [2].

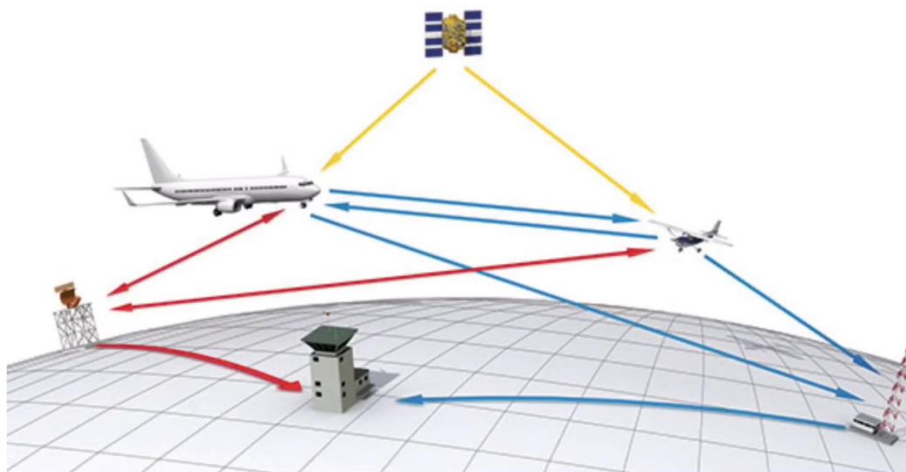


Рис.1. Схема ADS-B.

ADS-B (автоматическое зависящее наблюдение-вещание) – технология, позволяющая получать аэронавигационную информацию. Эта система использует транспондер в сочетании с GPS для передачи высокоточной информации о местоположении наземным диспетчерам, а также непосредственно на другие воздушные суда (рис. 1). Передача данных известна как ADS-B Out и ее точность выше, чем при обычном радиолокационном наблюдении. ADS-B поддерживает международный стандарт ICAO, который используется во всем мире.

ADS-B использует систему GPS для определения координат воздушного судна. Другие воздушные суда и наземные станции, оборудованные ADS-B, в радиусе порядка 150 миль принимают координаты воздушного судна, а также его тип, скорость, номер, рейс, курс, вертикальную скорость. Данная информация обновляется с интервалом времени 1 секунда по широкополосным каналам.

Организация запроса данных.

Организуем рабочий процесс в KNIME Analytics Platform для получения данных о воздушных судах в реальном времени (рис. 2). Api-запрос отправляется на Flightradar, в параметрах запроса указываются координаты двух точек, задающие область на карте в границах которой запрашивается информация.

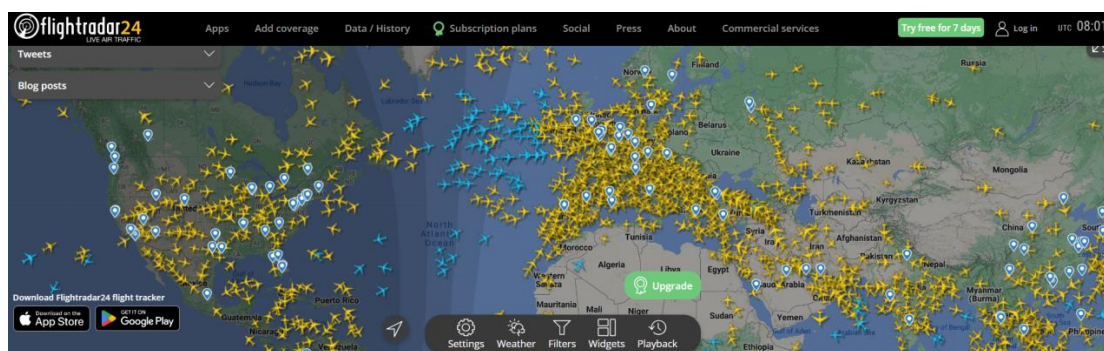


Рис. 2. Параметры запроса данных на портал.

Ресурс имеет ограничение по получению количества записей в одном api – до 1500 записей. По этой причине разобьем земной шар на 34 зоны для получения полных данных о всех воздушных судах в реальном времени.

В рабочий процесс в KNIME Analytics Platform добавим инструмент Table Creator, который позволяет создать таблицу данных. Введем параметры выделенных 34 зон в виде таблицы из 34 строк. Каждая строка включает параметры запроса с координатами северной широты и восточной долготы для заданных зон.

Используем инструмент Get Request для отправки GET запросов и получения нужных данных.

По умолчанию выходная таблица будет содержать столбец с полученными данными, типом их содержимого и кодом состояния HTTP. Узел пытается автоматически преобразовать полученные данные в тип данных KNIME на основе своего типа содержимого.

Для отправки запросов будем использовать столбец NEW URL состоящий из 34 строк. Следовательно, при запуске инструмента, последовательно будут отправляться 34 GET запроса (34 географические зоны) и записывать данные в новую колонку построчно. Инструмент добавляет еще одну колонку в таблицу с «ответными» данными в JSON формате.

В итоге, получаем порядка 1 миллиона записей, которые необходимо обработать.

Предварительная обработка данных.

Уберем лишние столбцы из таблицы с данными, оставив только столбец с JSON ответом. Для этого воспользуемся инструментом Column Filter (см. рисунок 3).

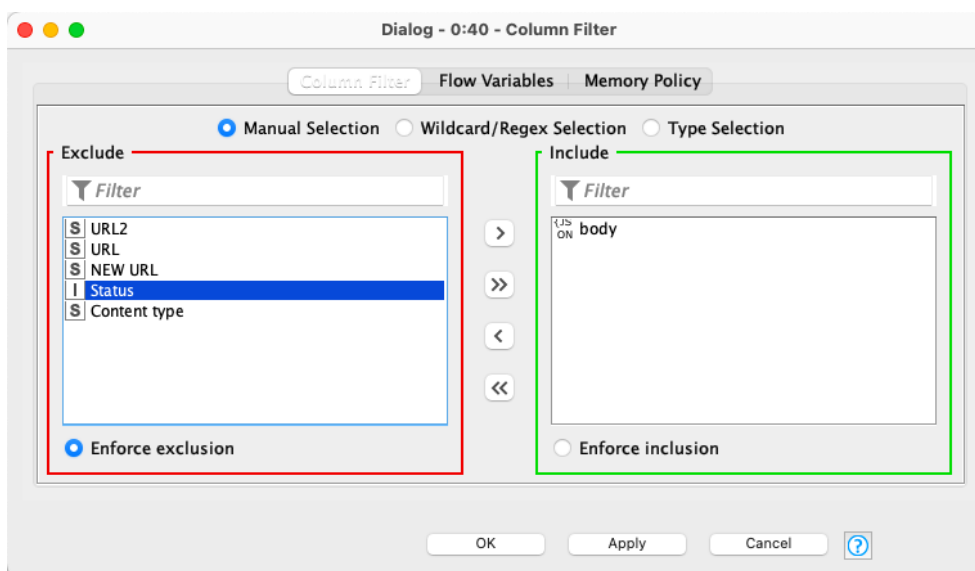


Рис. 3. Сортировка столбцов.

Рассмотрим подробнее данные которые «приходят» с сайта, для решений о дальнейшей выборке информации. Рассмотрим на примере одной из последних строк в первом из запросов.

```
1503 "2baa44c2":["155C33",57.36,20.11,196,34000,458,"F-EYSA4","SU95","RA-89139","PKV","KGD"]
```

Рис. 4. Пример массива данных одного из летательных аппаратов.

Каждая строка содержит массив данных (рис. 4):

"2baa44c2" – ключ строки в ответе, не несет смысловой нагрузки;

"155C33"– ICAO 24-bit ADDRESS (ICAO адрес ответчика с летательного аппарата);

"57.36"— широта;
 "20.11"— долгота;
 "196"— скорость в узлах;
 "3400"— высота в футах;
 "458"— курс;
 "F-EYSA4"— номер в системе;
 "SU95"— модель Самолета (Sukhoi Superjet 100);
 "RA-89139" – бортовой номер самолета;
 "PKV"— аэропорт вылета (Псков);
 "KGD"— аэропорт прилета (Калининград).

Для разбиения больших данных будем использовать инструмент JSON Path. JSON Path – это язык запросов для JSON, аналогичный XPath для XML (рис. 5).



Рис. 5. Фрагмент окна настроек узла JSON Path.

Для кластеризации данных, выберем нужные значения. ICAO ADDRESS используется для точного определения летательного аппарата и дальнейшего поиска информации о нем в системе. Так же будут взяты такие параметры как высота в футах, скорость в узлах, широта и долгота. Все численные данные, кроме адреса самолета будут типизированы как double.

При выборе 34 географических зон, для большей точности и недопущения пропуска какого-либо летательного аппарата, многие из зон были взяты с наложением друг на друга. В связи с этим возникает проблема дублирования данных. Так же во время выполнения запросов самолет может просто перелететь из одной зоны в другую что отобразится двойной записью данных для соседних зон. Для решения данной проблемы используем инструмент GroupBy. Он группирует строки таблицы по уникальным значениям в выбранных столбцах группы.

Будем искать дубликаты по двум параметрам, которые не меняются у летательных аппаратов и явно характеризуют их: это ICAO 24-BIT ADDRESS (String). Кроме этого, введем

дополнительный столбец SortColumn куда запишем бортовой номер судна (рис. 6). Дубликаты удаляются на основе этих параметров.

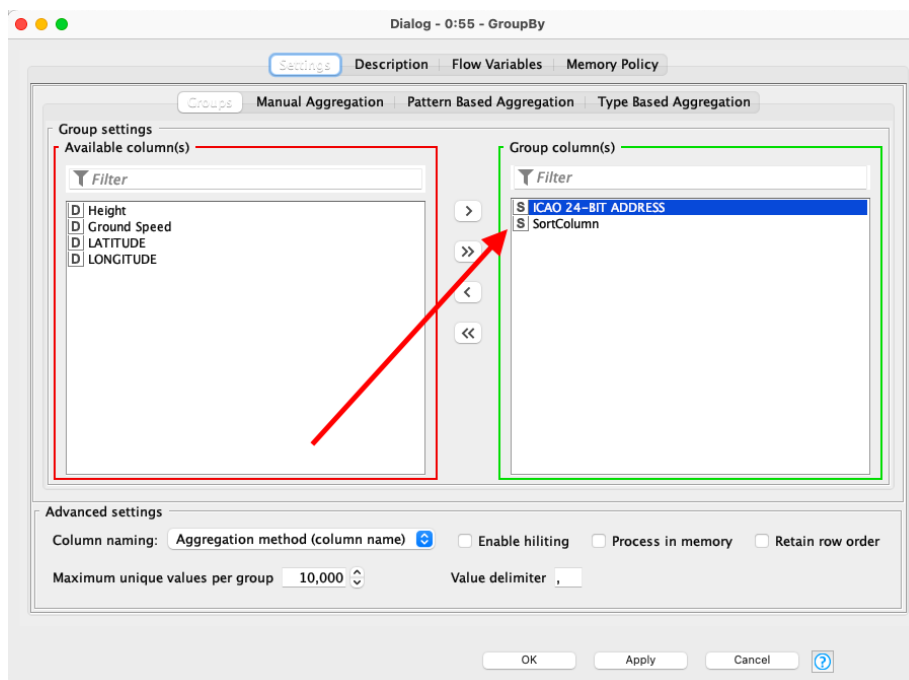


Рис. 6. Выбор критериев для поиска дублей.

Что бы визуализировать обработанную базу данных летательных аппаратов воспользуемся инструментом OSM Map View. Он не входит в стандартный пакет при установке KNIME. Загрузим его отдельно с открытого репозитория инструментов KNIME Hub, раздел OsmMapViewNodeFactory. Он предоставляет интерактивный вид карты мира. При желании на него также могут быть нанесены некоторые маркеры по данным широты и долготы.

На рисунке 7 по данным обработанной таблицы визуализированы все 13302 летательных аппарата (данные от 17 мая 2022 года).

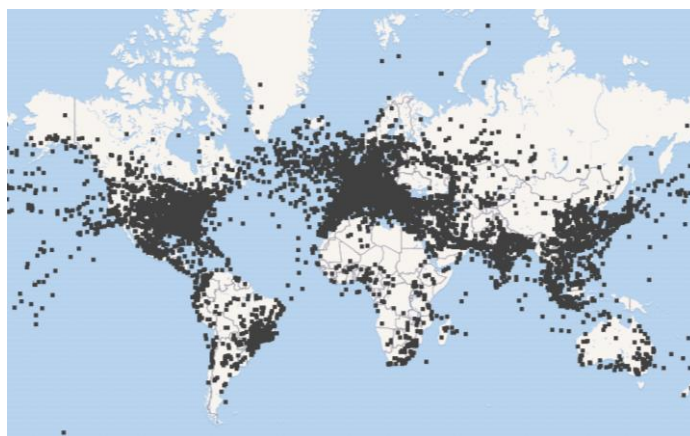


Рис. 7. Все летательные аппараты на карте мира.

Кластеризация летательных аппаратов по скорости и высоте.

Решим задачу кластеризации по двум параметрам: скорости и высоте. С помощью инструмента Column Filter были взяты из колонок таблицы, которые непосредственно нужны для данного примера кластеризации.

Выполним настройку узла кластеризации k-Means (рис. 8) исходя из проверки эффективности кластеризации методом силуэтов (инструмент Silhouette Coefficient) [1].

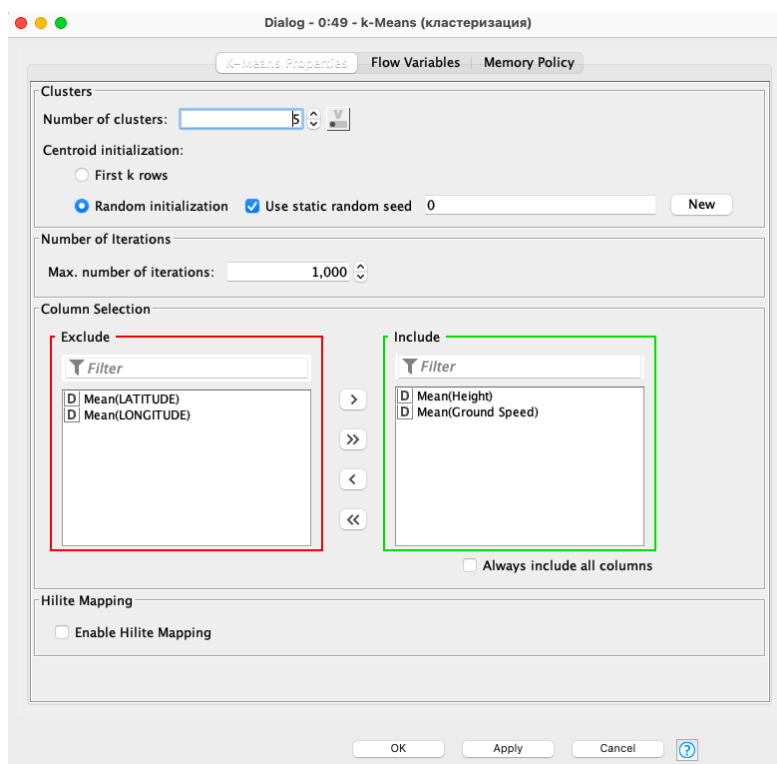


Рис. 8. Меню настроек k-Means.

Row ID	Mean ...
cluster_1	0.75
cluster_2	0.611
cluster_3	0.448
cluster_0	0.522
cluster_4	0.509
Overall	0.575

Рис. 9. Коэффициенты силуэтов.

Общее значение коэффициента силуэтов равно 0.575 (рис. 9).

Отсортируем таблицу по возрастанию кластеров и с помощью инструмента Color Manager окрасим точки каждого кластера в разные цвета.

Для вывода данных используем узел для построения точечных графиков Scatter Plot (рис. 10). Это точечная диаграмма с использованием библиотеки диаграмм на основе JavaScript. Доступ к представлению можно получить либо через действие «интерактивное представление» на исполняемом узле, либо на странице веб-портала KNIME Server.

По оси абсцисс отмечена скорость в узлах, по оси ординат высота в футах.

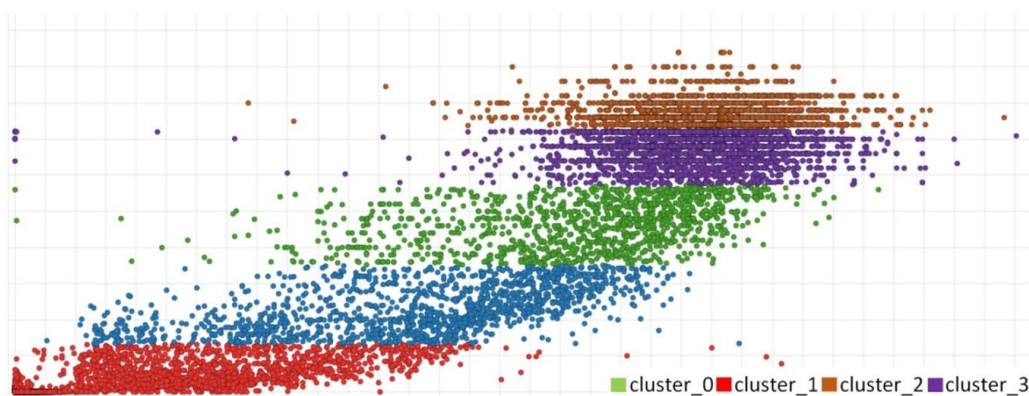


Рис. 10. Распределение данных по кластерам.

Интерпретация результатов

Проведем соответствие каждого кластера техническим данным полетов, классу и состоянию воздушного судна.

Рассмотрим кластер 1 (cluster_1, красный цвет). Здесь наблюдаем относительно небольшую скорость и высоту летательных аппаратов. Так же есть скопление точек около нулевых отметок. Это самолеты, которые совершили посадку и стоят на земле с нулевой высотой и скоростью. Точки, которые имеют нулевую высоту, но уже возросшую скорость, от 20 до 150 узлов можно соотнести с самолетами, которые совершают руление по аэродрому, а также идут на взлет по полосе. Это подтверждается техническим данным – скорость руления 30 узлов, скорость взлета среднемагистральных самолетов (Boeing 737) 145 узлов. Точки, имеющие небольшую высоту и скорость соответствуют самолетам, которые взлетели и набирают крейсерскую высоту или идут на посадку. Таким образом, кластеру 1 (cluster_1, красный цвет) соответствуют самолеты, которые находятся на земле, взлетающие или совершающие посадку.

Рассмотрим кластер 2 (cluster_4). Здесь разброс высот составляет примерно от 6 тыс. до 15 тыс. футов. А скорость варьируется от 60 до 290 узлов. Этому кластеру можно отнести

вертолёты, высота полета которые соответствует данным, а скорость невелика. Так же к этому кластеру отнесем легкомоторные самолеты, такие как АН-2 (кукурузник) или Cessna 172 (самый популярный легкомоторный самолет в мире). Их параметры полета по высоте и скорости соответствуют данным значениям.

К кластеру 3 (cluster_0, зеленый цвет) соответствуют высоты от 20 тыс. до 30 тыс. футов. В этот кластер можем отнести ближнемагистральные и среднемагистральные самолеты гражданской авиации.

К кластеру 4 и 5 (cluster_3, cluster_2) будут соответствовать воздушные суда, проходящие на больших эшелонах. Это дальнемагистральные и межконтинентальные пассажирские и грузовые самолеты.

Также наблюдаем несколько точек выброса. Это такие летательные аппараты, которые имеют нулевую скорость и большую высоту или очень низкую высоту и запредельную скорость. Скорее всего такие данные свидетельствуют о неисправности оборудования или возникновении аварийных ситуаций. Все исследованные точки выброса на данных от 17 мая 2022 года 15:00 МСК, оказались технической неисправностью оборудования со стороны приемника. Также в точку выбросов, предположительно, попали два военных истребителя, которые работали на малых высотах с большой скоростью.

Для наглядности, с помощью инструмента OSM Map View, были выведены на карту мира все летательные аппараты с цветами точек соответствующих кластерам (рис. 11).

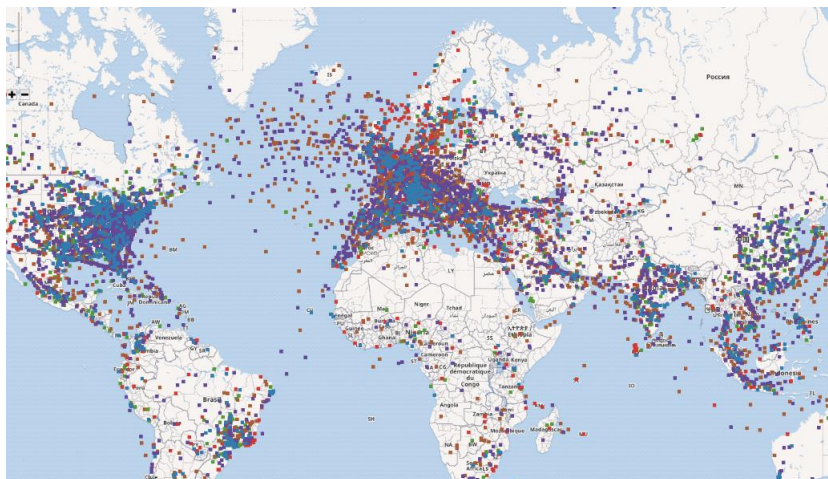


Рис. 11. Летательные аппараты с цветами точек по кластерам.

Также на карту выведем кластер 1 – самолеты, которые находятся на земле, взлетающие или совершающие посадку. Скобление таких самолетов происходит вблизи больших городов и аэропортов (рис. 12).



Рис. 12. Самолеты, которые находятся на земле, взлетающие или совершающие посадку.

При выводе на карту кластера 2 – это воздушные судна, проходящие на больших эшелонах, дальнемагистральные и межконтинентальные самолеты. Можем наблюдать линии высотных воздушных трасс (рис. 13).

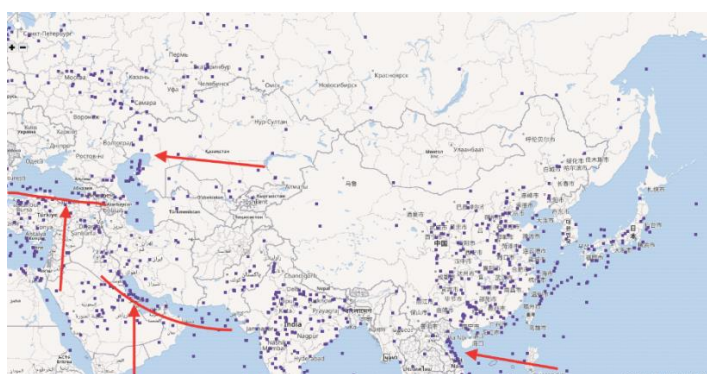


Рис.13. Линии воздушных трасс.

Таким образом, в KNIME Analytics Platform был построен рабочий процесс, который может представлять практическую ценность при решении задач визуализации воздушных судов в режиме реального времени.

СПИСОК ЛИТЕРАТУРЫ

1. Гарин М. А., Егорова Д. К., Сайфетдинов С. Ф. Создание workflow аналитической платформы KNIME для анализа данных на примере вакансий сайта HeadHunter [Электронный ресурс] // Огарев-online. Раздел «Физико-математические науки». – 2021. – №12. – Режим доступа: <http://journal.mrsu.ru/arts/sozdanie-workflow-analiticheskoy-platformy-knime-dlya-analiza-dannyx-na-primere-vakansij-sajta-headhunter> (дата обращения: 20.11.2021).
2. Flightradar24: Live Flight Tracker Map [Электронный ресурс]. – Режим доступа: <https://www.flightradar24.com/54.18.45.18/6> (дата обращения: 15.05.2022).
3. KNIME Analytics Platform Map [Электронный ресурс]. – Режим доступа: <https://www.KNIME.com/KNIME-analytics-platform> (дата обращения 10.05.2021).

РАССАДИН А. Э.

ТОЧНОЕ РЕШЕНИЕ ОДНОГО ФУНКЦИОНАЛЬНО-ДИФФЕРЕНЦИАЛЬНОГО
УРАВНЕНИЯ ПАРАБОЛИЧЕСКОГО ТИПА
С ПОМОЩЬЮ ТЕОРИИ ПОЛУГРУПП И НЕКОТОРЫЕ ЕГО ПРИМЕНЕНИЯ

Аннотация. В работе построен оператор C_0 -полугруппы, инфинитезимальный генератор которой является линейной комбинацией оператора диффузии-теплопроводности и оператора пространственной инверсии на прямой. Обсуждены общие свойства эволюции функций, задающих начальные условия, под действием такого оператора. Показано, что эти свойства резко отличаются от свойств решений одномерного параболического уравнения на прямой за счёт наличия по сути дискретного элемента в генераторе рассматриваемой полугруппы.

Ключевые слова: коммутатор, проектор, пространство Лебега, гиперболические функции, преобразование Лапласа, интегральное ядро оператора, резольвента.

RASSADIN A. E.

EXACT SOLUTION TO ONE FUNCTIONAL DIFFERENTIAL EQUATION
OF PARABOLIC TYPE USING SEMIGROUP THEORY
AND SOME OF ITS APPLICATIONS

Abstract. In this paper, the C_0 -semigroup operator is constructed, the infinitesimal generator of which is a linear combination of the diffusion-thermal conductivity operator and the spatial inversion operator on a straight line. The general properties of the evolution of functions defining initial conditions under the action of such an operator are discussed. It is shown that these properties differ sharply from the properties of solutions to a one-dimensional parabolic equation on a straight line due to the presence of an essentially discrete element in the generator of the semigroup under consideration.

Keywords: commutator, projector, Lebesgue space, hyperbolic functions, Laplace transform, integral operator kernel, resolvent.

Функционально-дифференциальные уравнения параболического типа активно используются при решении целого ряда задач оптоэлектроники [1] и математической биологии [2]. Многие общие свойства уравнений такого типа описаны в обзоре [3]. Однако в математической физике важную роль играют не только свойства решений её уравнений, но и построение точных решений этих последних, поэтому рассмотрим задачу Коши на прямой:

$$\frac{\partial u(x,t)}{\partial t} = \frac{\partial^2 u(x,t)}{\partial x^2} + \alpha \cdot u(-x,t), \quad u(x,0) = u_0(x), \quad x \in R, \quad \alpha \neq 0. \quad (1)$$

Для этой задачи справедлива следующая теорема.

Теорема. Пусть $u_0 \in L_\infty(R)$, тогда точное решение задачи Коши (1) имеет вид:

$$u(x, t) = \int_{-\infty}^{+\infty} g(x, \xi; t) \cdot u_0(\xi) \cdot d\xi, \quad (2)$$

где интегральное ядро оператора эволюции равно

$$g(x, \xi; t) = ch(\alpha \cdot t) \cdot G(x - \xi; t) + sh(\alpha \cdot t) \cdot G(x + \xi; t), \quad (3)$$

а

$$G(x; t) = \frac{1}{2 \cdot \sqrt{\pi \cdot t}} \cdot \exp\left(-\frac{x^2}{4 \cdot t}\right). \quad (4)$$

Доказательство. Перепишем задачу Коши (1) для исходного функционально-дифференциального уравнения параболического типа в операторном виде:

$$\frac{du}{dt} = Au, \quad u(0) = u_0, \quad (5)$$

где

$$A = D^2 + \alpha \cdot P \quad (6)$$

— линейный оператор, действие которого на произвольную функцию $f(x)$ можно определить с помощью следующих правил: $(Df)(x) = f'(x)$ (оператор дифференцирования) и $(Pf)(x) = f(-x)$.

Как хорошо известно [4], решение абстрактной задачи Коши (5) равно:

$$u(t) = \exp(At)u_0. \quad (7)$$

Далее, входящий в формулу (6) оператор P есть не что иное, как широко используемый в квантовой механике оператор пространственной инверсии [5].

Этот оператор коммутирует с квадратом оператора дифференцирования D : $[D^2, P] = 0$, следовательно:

$$\exp[(D^2 + \alpha P) \cdot t] = \exp(\alpha \cdot t \cdot P) \cdot \exp(t \cdot D^2). \quad (8)$$

Легко убедиться в том, что в пространстве $L_\infty(R)$ оператор инверсии P ограничен, причём $\|P\| = 1$, поэтому:

$$\exp(\alpha \cdot t \cdot P) = \sum_{n=0}^{\infty} \frac{\alpha^n t^n}{n!} \cdot P^n. \quad (9)$$

Но так как $P^2 = I$, где I — тождественный оператор [5], то формула (9) упрощается:

$$\exp(\alpha \cdot t \cdot P) = ch(\alpha \cdot t) \cdot I + sh(\alpha \cdot t) \cdot P. \quad (10)$$

Наконец, $\{\exp(t \cdot D^2)\}_{t \geq 0}$ — оператор полугруппы Гаусса-Вейерштрасса, действие которого на произвольную функцию $f \in L_\infty(R)$ задаётся её свёрткой с функцией (4) [6]:

$$(\exp(t \cdot D^2) \cdot f)(x) = \int_{-\infty}^{+\infty} G(x - \xi; t) \cdot f(\xi) \cdot d\xi. \quad (11)$$

Комбинируя формулы (7), (8), (10) и (11), придём к выражениям (2) и (3), то есть к условию теоремы.

Теорема полностью доказана.

Интегрируя соотношение (2) по x по всей прямой и переставляя затем порядки интегрирования по x и по ξ , получим следующее утверждение.

Следствие 1. Если $u(x, t)$ — точное решение задачи Коши (1) с начальным условием $u_0 \in L_1(R)$, то:

$$\int_{-\infty}^{+\infty} u(x, t) \cdot dx = \exp(\alpha \cdot t) \cdot \int_{-\infty}^{+\infty} u_0(x) \cdot dx. \quad (12)$$

Соотношение (12) демонстрирует разницу между поведением решений обычного уравнения параболического типа ($\alpha = 0$) и функционально-дифференциального уравнения параболического типа ($\alpha \neq 0$), а именно, площадь, ограниченная решением уравнения (1) и осью x , со временем либо экспоненциально растёт (при $\alpha > 0$), либо экспоненциально убывает (при $\alpha < 0$) — в отличие от площади, ограниченной решением уравнения диффузии-теплопроводности ($\alpha = 0$), которая не меняется со временем.

Переходя в формуле (10) от гиперболических функций к экспоненциальным и используя затем формулу (8), легко вывести ещё одно следствие из доказанной выше теоремы.

Следствие 2. Оператор полугруппы, соответствующий инфинитезимальному генератору полугруппы (6), можно представить в виде:

$$\exp(At) = [\exp(\alpha \cdot t) \cdot P_+ + \exp(-\alpha \cdot t) \cdot P_-] \cdot \exp(t \cdot D^2), \quad (13)$$

где $P_\pm = (I \pm P)/2$ — операторы, удовлетворяющие соотношениям $P_\pm^2 = P_\pm$, то есть проекционные операторы.

Это представление эволюционного оператора задачи Коши (1) позволяет лучше понять общую структуру её точного решения (2), а именно, действуя согласно формуле (7) на начальное условие $u_0(x)$ оператором полугруппы в представлении (13), получим, что:

$$u(x, t) = \exp(\alpha \cdot t) \cdot u^+(x, t) + \exp(-\alpha \cdot t) \cdot u^-(x, t), \quad (14)$$

где $u^\pm(x,t) = (u^0(x,t) \pm u^0(-x,t))/2$ — соответственно чётная и нечётная части от функции $u^0(x,t) = (\exp(t \cdot D^2)u_0)(x)$, являющейся решением обычного уравнения теплопроводности с тем же начальным условием $u_0(x)$. Таким образом, из выражения (14) следует, что на больших временах при $\alpha > 0$ остаётся и экспоненциально растёт чётная часть $u^+(x,t)$ от функции $u^0(x,t)$, а при $\alpha < 0$ остаётся и экспоненциально растёт нечётная часть $u^-(x,t)$ от функции $u^0(x,t)$.

Важной характеристикой инфинитезимального генератора (6) рассматриваемой полугруппы является его резольвента $R_\lambda(A) = (A - \lambda I)^{-1}$ [4, 6].

Пусть функция $r(x, \xi; \lambda)$ — интегральное ядро оператора резольвенты. Оно действует на произвольный вектор $f \in L_\infty(R)$ следующим образом:

$$(R_\lambda(A)f)(x) = \int_{-\infty}^{+\infty} r(x, \xi; \lambda) \cdot f(\xi) \cdot d\xi. \quad (15)$$

Применив к выражениям (2)-(4) формулу связи между оператором резольвенты и оператором полугруппы [4, 6]:

$$(R_\lambda(A)f)(x) = - \int_0^{+\infty} \exp(-\lambda \cdot t) \cdot (\exp(At)f)(x) \cdot dt \quad (16)$$

и сравнив между собой выражения (15) и (16) после использования известных свойств преобразования Лапласа [7] и известного выражения интегрального ядра для резольвенты $R_\lambda(D^2)$ оператора D^2 [6], несложно получить ещё одно утверждение.

Следствие 3. *Интегральное ядро оператора резольвенты инфинитезимального генератора полугруппы (6) равно:*

$$r(x, \xi; \lambda) = \frac{\rho(x, \xi; \lambda - \alpha) + \rho(x, -\xi; \lambda - \alpha) + \rho(x, \xi; \lambda + \alpha) - \rho(x, -\xi; \lambda + \alpha)}{2}, \quad (17)$$

где $\rho(x, \xi; \lambda) = -\exp(-\sqrt{\lambda} \cdot |x - \xi|) / (2 \cdot \sqrt{\lambda})$ — интегральное ядро резольвенты $R_\lambda(D^2)$.

С другой стороны, подстановка представления (13) оператора рассматриваемой полугруппы в формулу (16) позволяет переписать выражение для резольвенты в операторном виде:

$$R_\lambda(D^2 + \alpha \cdot P) = P_+ \cdot R_{\lambda-\alpha}(D^2) + P_- \cdot R_{\lambda+\alpha}(D^2). \quad (18)$$

Очевидно, что формулы (17) и (18) эквивалентны друг другу.

Рассмотрим пример точного решения задачи Коши (1). Выберем её начальное условие в виде гауссоиды с характерной шириной l_0 , сдвинутой относительно начала координат на расстояние x_0 :

$$u_0(x) = \exp\left[-\frac{(x-x_0)^2}{4 \cdot l_0^2}\right]. \quad (19)$$

Оператор полугруппы Гаусса-Вейерштрасса $(\exp(D^2 t))_{t \geq 0}$ трансформирует во времени функцию (19) следующим образом:

$$u^0(x, t) = \frac{l_0}{\sqrt{l_0^2 + t}} \exp\left[-\frac{1}{4} \frac{(x-x_0)^2}{l_0^2 + t}\right]. \quad (20)$$

Наконец, из функции (20) по формуле (14) получаем точное решение задачи Коши (1) с начальным условием (19):

$$u(x, t) = \frac{l_0}{\sqrt{l_0^2 + t}} \cdot \left(ch(\alpha t) \cdot \exp\left[-\frac{1}{4} \frac{(x-x_0)^2}{l_0^2 + t}\right] + sh(\alpha t) \cdot \exp\left[-\frac{1}{4} \frac{(x+x_0)^2}{l_0^2 + t}\right] \right). \quad (21)$$

Графики функции (21) при $x_0 = 2$, $l_0 = 1$ и различных значениях α приведены на рис. 1 и 2.

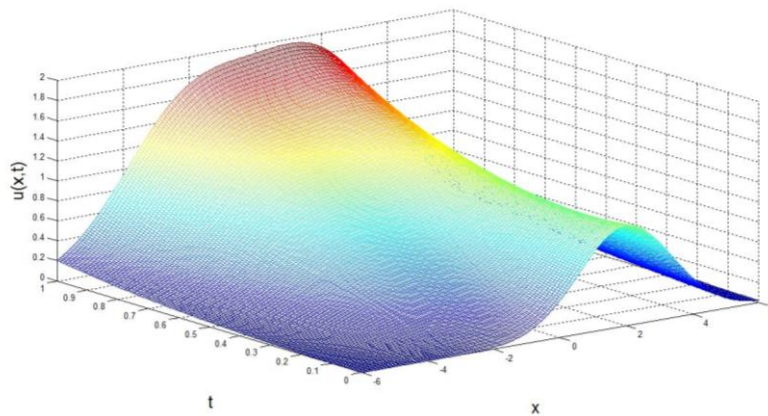


Рис. 1. Временная эволюция смещённой гауссоиды при $\alpha = 1,5$.

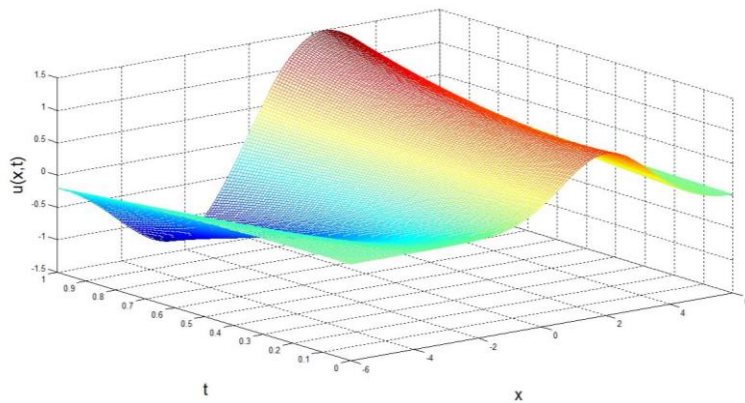


Рис. 2. Временная эволюция смещённой гауссоиды при $\alpha = -1,5$.

Из рисунка 1 видно, что на больших временах экспоненциально растёт чётная часть гауссоиды (20). Из рисунка 2 видно, что на больших временах экспоненциально растёт её нечётная часть. Таким образом, этот пример полностью иллюстрирует доказанную в данной работе теорему.

В заключение необходимо отметить, что полученные результаты могут быть применены для дальнейшего развития теории полугрупп, в частности, построенное точное решение задачи Коши (1) может быть положено в основу различных схем теории возмущений, а найденное в следствии 3 теоремы интегральное ядро оператора резольвенты инфинитезимального генератора полугруппы (6) даёт возможность вычислить резольвенты последовательных степеней этого оператора.

СПИСОК ЛИТЕРАТУРЫ

1. Skubachevskii A. L. Bifurcation of periodic solutions for nonlinear parabolic functional differential equations arising in optoelectronics // *Nonlinear Analysis: Theory, Methods & Applications*. – 1998. – V. 32, № 2. – P. 261–278.
2. Yoshida K. The Hopf bifurcation and its stability for semilinear diffusion equations with time delay arising in ecology // *Hiroshima Mathematical Journal*. – 1982. – V. 12, № 2. – P. 321–348.
3. Муравник А. Б. Функционально-дифференциальные параболические уравнения: интегральные представления и качественные свойства решений задачи Коши // *Современная математика. Фундаментальные направления*. – 2014. – Т. 52. – С. 3–141.
4. Крейн С. Г. Линейные дифференциальные уравнения в банаховом пространстве. – М.: Наука, 1967. – 464 с.
5. Ландау Л. Д., Лифшиц Е. М. Квантовая механика (нерелятивистская теория). – М.: Наука, 1989. – 768 с.
6. Хилле Э., Филлипс Р. С. Функциональный анализ и полугруппы. – М.: ИЛ, 1962. – 830 с.
7. Диткин В. А., Прудников А. П. Интегральные преобразования и операционное исчисление. – М.: ГИФМЛ, 1961. – 524 с.

БУТКИНА А. А., КОЧКАЕВА Ю. С., ЧУДНОВА Р. А.

**РАЗРАБОТКА ОСНОВНОГО ФУНКЦИОНАЛА ИГРОВОГО ПРИЛОЖЕНИЯ ДЛЯ
ПЕРСОНАЛЬНЫХ КОМПЬЮТЕРОВ С ИСПОЛЬЗОВАНИЕМ UNREAL ENGINE**

Аннотация. В статье описана реализация авторами основного функционала, разработанного на игровом движке «Unreal Engine» тестового приложения, предназначенного для запуска на персональных компьютерах. Разработка выполнялась с использованием визуального блокового языка программирования «Blueprint».

Ключевые слова: игровой движок, Unreal Engine 4, функциональные возможности, скрипт, визуальный блоковый язык программирования Blueprint.

BUTKINA A. A., KOCHKAREVA YU. S., CHUDNOVA R. A.

**DEVELOPMENT OF THE MAIN FUNCTIONALITY OF GAMING APPLICATION
FOR PERSONAL COMPUTERS USING UNREAL ENGINE**

Abstract. The article describes the authors' implementation of the basic functionality of a test application developed on the game engine Unreal Engine and designed to run on personal computers. The development was performed using the node-based visual programming language Blueprint.

Keywords: game engine, Unreal Engine 4, functionality, script, node-based visual programming language Blueprint.

Введение. В настоящее время наблюдается всё возрастающий интерес разработчиков программного обеспечения к сфере разработки компьютерных игр и других интерактивных развлечений. Главным образом это связано с возросшей популярностью игр во время пандемии в 2020 году, когда игры стали своего рода альтернативой другим видам досуга. Однако многие аналитические источники [1] утверждают, что тенденция к росту популярности видеоигр будет сохраняться и в дальнейшем. Так, в 2021 году объем российского рынка видеоигр достиг 177,4 млрд рублей, показав увеличение по сравнению с соответствующими итогами 2020 года на 9% [2]. Поэтому среди разработчиков приложений всё большую актуальность приобретает изучение основных принципов работы в игровых движках, а также создание приложений в них с применением языков визуального программирования.

Таким образом, данное исследование, посвященное рассмотрению реализации тестового игрового приложения, предназначенного для персональных компьютеров, на визуальном языке программирования с применением в качестве среды разработки игрового движка является актуальным.

Для достижения этой цели были поставлены и решены следующие задачи:

- выполнить анализ наиболее популярных игровых движков и на его основе сделать выбор оптимальной среды разработки;
- рассмотреть особенности использования языков визуального программирования;
- реализовать основные функции разрабатываемого приложения.

Анализ игровых движков. Игровой движок – особый класс программного обеспечения, который используется в качестве среды разработки игровых приложений. Современные разработчики видеоигр чаще всего используют следующие игровые движки: CryEngine, Unity, Unreal Engine. Каждый из них имеет свои особенности, но в тоже время существуют и общие черты, которые позволяют создавать проекты высокого уровня. Авторами были изучены основные принципы работы в каждой из данных сред разработки, а также проведён их сравнительный анализ, основные результаты которого приведены в таблице 1.

Таблица 1

Сравнение игровых движков

	CryEngine	Unity	Unreal Engine
Лицензия	Условно- бесплатная	Условно- бесплатная	Условно-бесплатная
Язык программирования	C++	C#	C++
Визуальная система программирования	Flow Graph	Bolt – требуется покупка	Blueprint
Внедрение 3D моделей в движок	+	+	+
Библиотека ассетов	Маленькая	Большая	Большая
Оптимизация	+	-	+/-
Кроссплатформенность	Windows, PS, XBOX	Windows, IOS, Android	Windows, IOS, Android, PS, XBOX
Тестирование в реальном времени	+	+	+
Дополнительное ПО для работы	-	+	-
Учебные пособия	-	+	+
Язык интерфейса	Английский	Английский	Английский
Техническая поддержка	-	+	+

Таким образом, в результате проведенного сравнительного анализа в качестве наиболее подходящего для разработки тестового приложения был выбран игровой движок «Unreal Engine», являющийся безоговорочным лидером среди всех аналогов.

Unreal Engine. В частности, «Unreal Engine» был выбран авторами благодаря своей доступности (является условно-бесплатным) и большим количеством возможностей, за которые в других движках пришлось бы доплачивать, например, за визуальный блоковый язык программирования «Blueprint», который облегчает и ускоряет процесс написания кода и не требует особых знаний C++, языка программирования, поддерживаемого для написания скриптов в «Unreal Engine». Кроме того, данный движок имеет достаточно большое количество учебных пособий и возможность обратиться в техподдержку по решению возникшей проблемы или за получением дополнительной информации [3].

Следует учитывать, что для полноценной стабильной работы в данной среде разработки на компьютере необходимо выполнение следующих системных требований для операционных систем семейства Windows [4]:

- операционная система: Windows 10 64-bit;
- процессор: Intel или AMD 4 ядра и более, тактовая частота 2,5 ГГц или выше;
- оперативная память: 8 ГБ;
- видеокарта: совместимая с DirectX 11 или DirectX 12;
- 22 ГБ свободного места на жёстком диске.

Для написания функциональных скриптов программа «Unreal Engine» поддерживает два языка программирования: «C++» и «Blueprint». Для разработки программы авторами было принято решение использовать язык программирования «Blueprint».

Blueprint. «Blueprint» – строго типизированный визуальный скриптовый язык, с помощью которого можно описать логику приложения любой сложности без использования языков программирования. Другими словами, это тот же C++, но в более удобном блочном виде [5]. Главным плюсом такого языка является простота при его использовании. Вместо написания кода предлагается возможность составлять системы из логических блоков. С помощью такой системы можно «собрать» программу любой сложности, главное – понимать основные принципы объектно-ориентированного программирования [6].

Однако, у данного языка программирования также есть и недостатки. В частности, одним из его недостатков является громоздкость, т.к. при составлении простого действия необходимо использование нескольких блоков, а при наличии большого количества операций система быстро становится слишком нагруженной и трудной для восприятия.

Реализованный функционал приложения. Авторами были реализованы следующие основные функциональные возможности разрабатываемого приложения, позволяющие пользователю осуществлять управление главным персонажем.

- 1) Управление игровым персонажем

В первую очередь перед созданием логики передвижения персонажа в игровом движке были определены и настроены клавиши, которые при нажатии вызывают определённое действие (рис. 1).

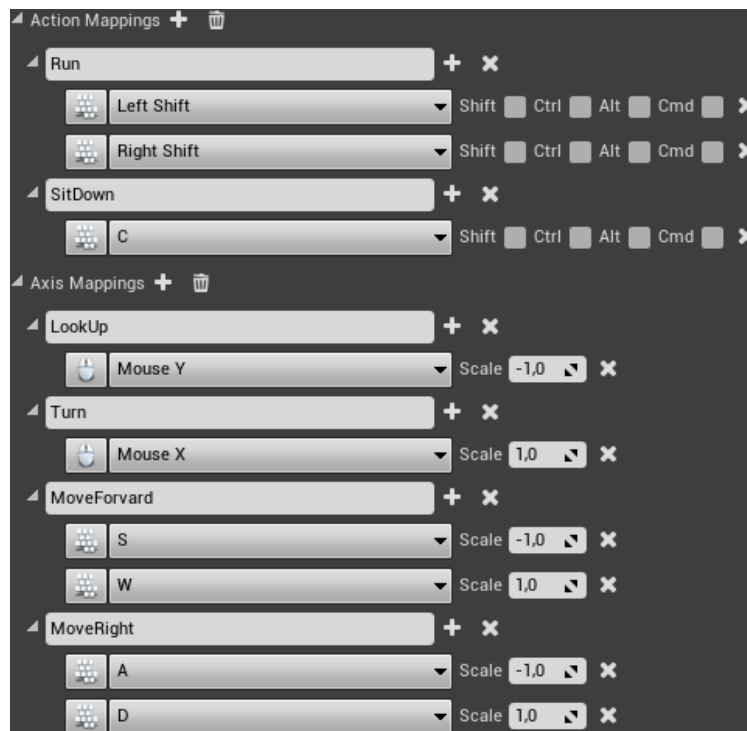


Рис. 1. Выбор и настройка клавиш.

Таким образом, персонаж будет передвигаться вперёд/назад при нажатии на клавиши «W»/«S», а также вправо/влево при нажатии на клавиши «D»/«A». Считывание сигнала с клавиатуры реализовано следующим образом (рис. 2).

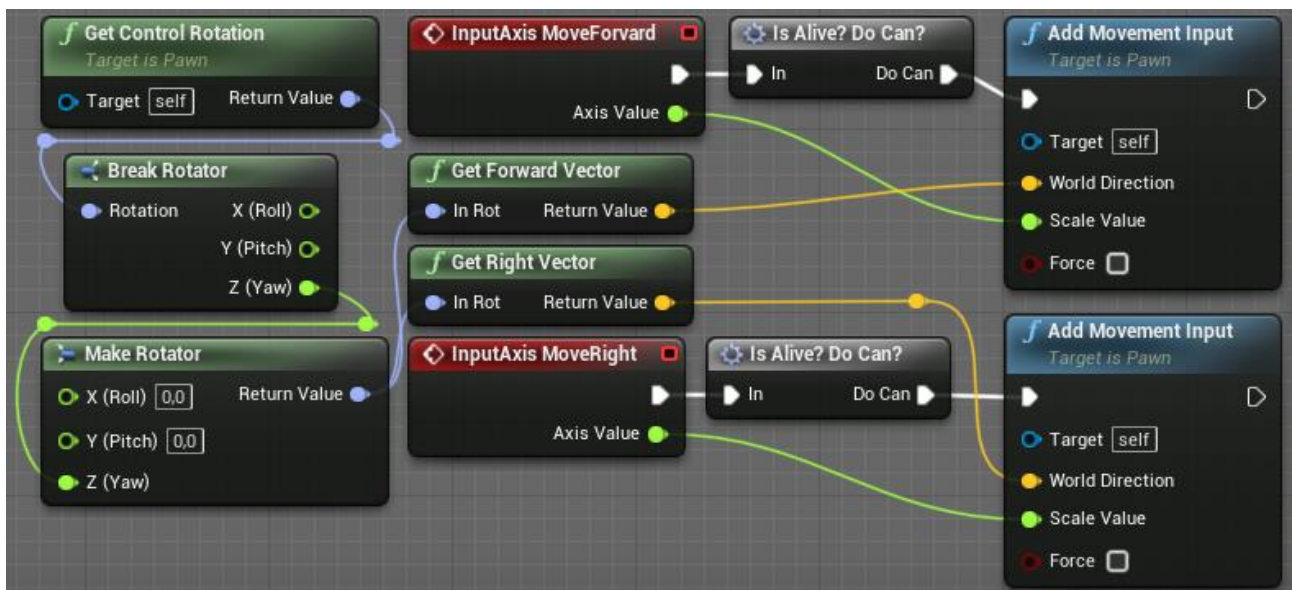


Рис. 2. Модуль логики управления главным персонажем.

Для более реалистичного игрового процесса была добавлена возможность вращения камерой с помощью мыши. Вектор X отвечает за поворот влево/вправо, вектор Y - за поворот вверх/вниз. В данном случае нам нужно получать текущие значения осей мыши и вращать камеру в зависимости от её движения (рис. 3).



Рис. 3. Модуль логики управления камерой.

Для бега и передвижения персонажа в положении «полуприсед» авторами были реализованы отдельные модули логики (рис. 4, 5). Основная логика построения блоковой системы для функции «бег» состоит в следующем: изначально необходимо определить, какова скорость персонажа. Далее, при нажатии и удерживании любой клавиши Shift, задаётся значение скорости бега/шага персонажа путём прибавления/вычитания ускорения к полученному на предыдущем шаге результату. Передвижение в положении «полуприсед» происходит после получения сигнала о смене положения персонажа, т.е. после нажатия на клавишу С, при этом скорость передвижения самого персонажа становится медленнее.

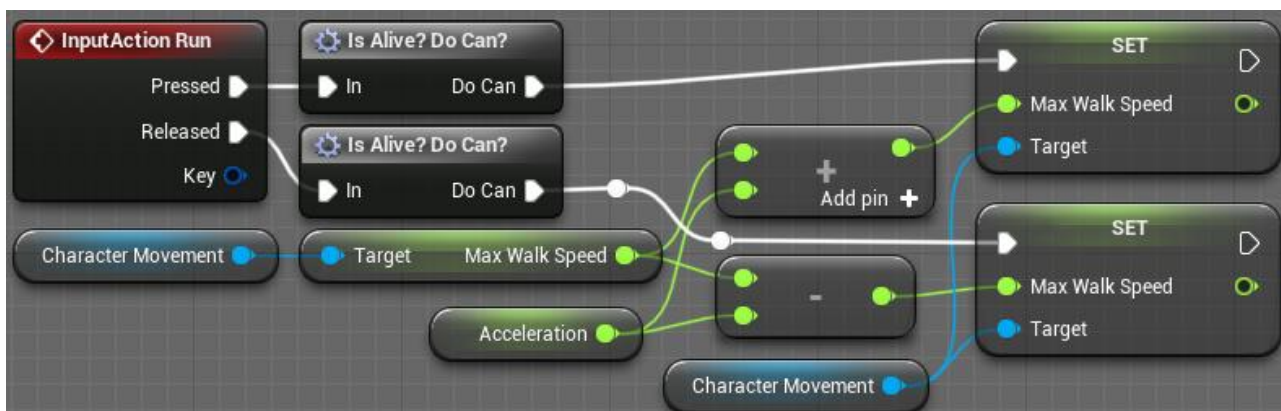


Рис. 4. Модуль логики, отвечающий за быстрое передвижение персонажа.

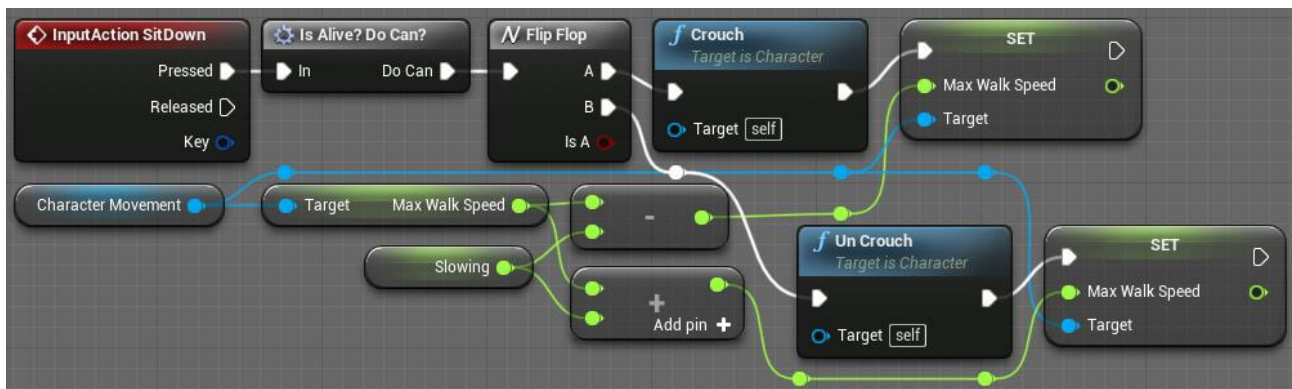


Рис. 5. Модуль логики, отвечающий за передвижение персонажа в положении «полуприсед».

2) Виджет шкал жизненных показателей

Для визуального представления специальных численных показателей «Выносливость» и «Рассудок», реализуемых в тестовом игровом приложении, авторами используется виджет. Виджет (widget) представляет собой элемент графического интерфейса пользователя, используемый в программе «Unreal Engine» для создания человеко-машинного интерфейса программы. Для этого первоначально создаётся дизайн виджета, в который добавляются все необходимые компоненты из встроенных в игровой движок элементов, например, «Полоса прогресса», «Текстовое поле» и т.п. (рис. 6). Далее с помощью языка «Blueprint» реализуется логика работы виджета, т.е. добавляются отклики на определённые действия (рис. 7–9). На заключительном этапе устанавливается событие, позволяющее игроку открыть определённый виджет в процессе работы приложения (рис. 10).

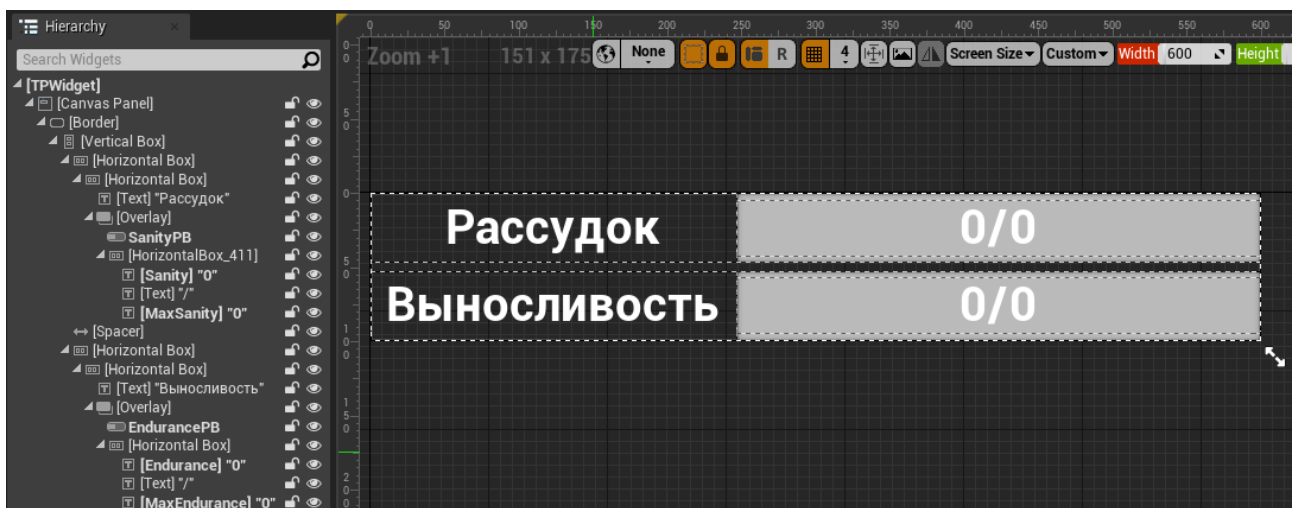


Рис. 6. Создание интерфейса виджета «Шкалы жизненных показателей».

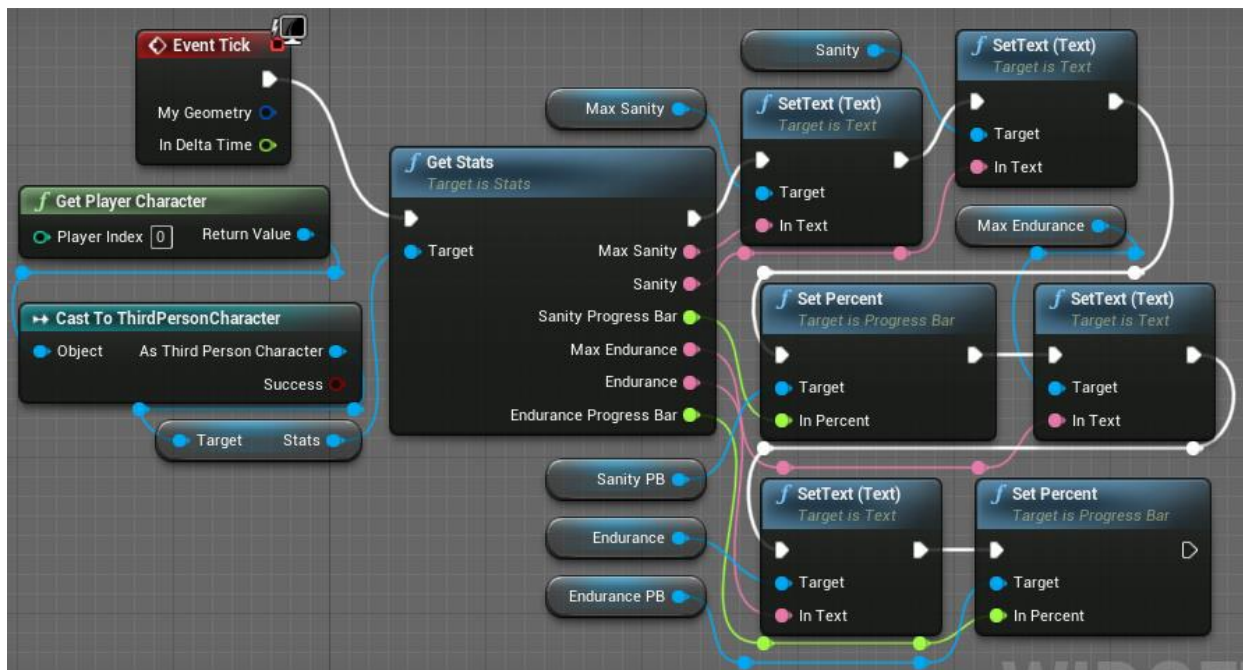


Рис. 7. Модуль логики передачи значений жизненных показателей в виджете «Шкалы жизненных показателей».

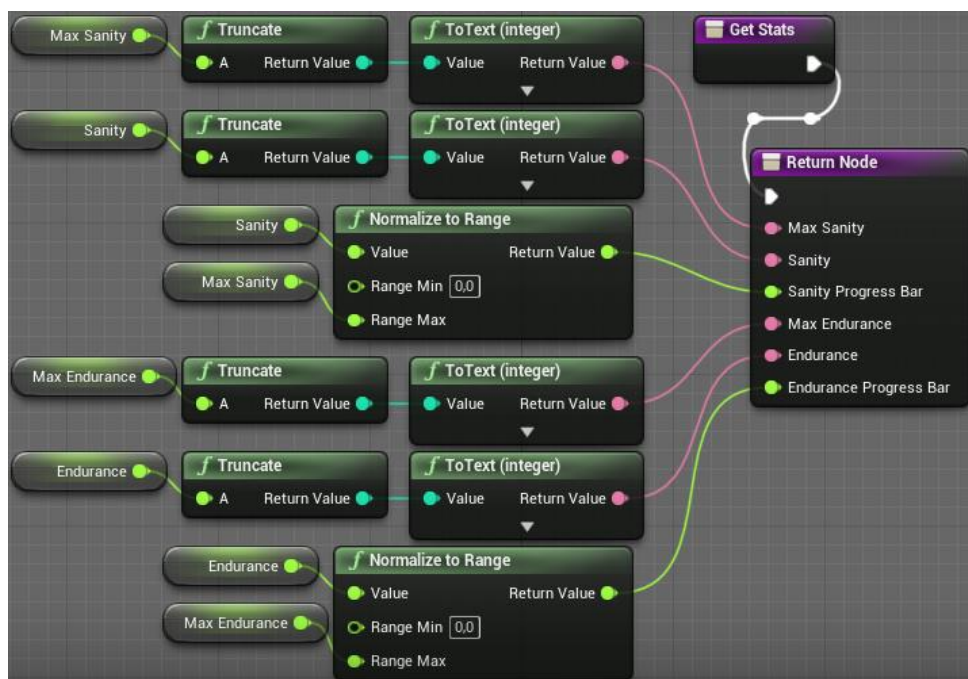


Рис. 8. Модуль логики получения значений жизненных показателей.

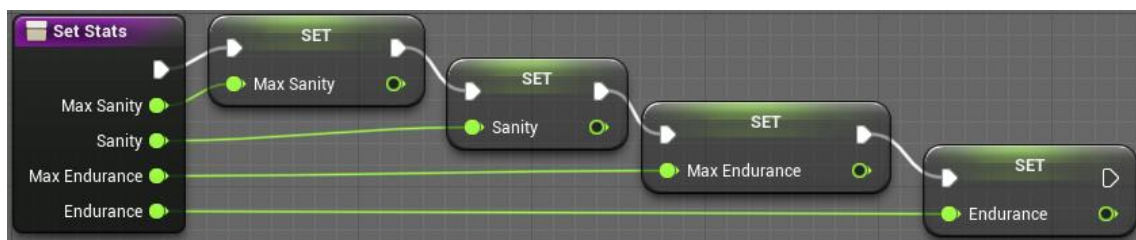


Рис. 9. Модуль логики записи значений жизненных показателей.



Рис. 10. Настраиваемое событие для виджета «Шкалы жизненных показателей».

3) Отображение жизненных показателей главного персонажа

Во многих играх выносливость персонажа зависит от скорости прохождения уровней, причём под скоростью подразумевается скорость шага/бега персонажа. Так в разрабатываемом тестовом приложении при передвижении персонажа в режиме «бег» теряется выносливость. Когда её количество достигает отметки «0», персонаж перестаёт передвигаться в режиме «бег». Количество выносливости автоматически восстанавливается при условии, если персонаж передвигается обычным шагом (рис. 11).

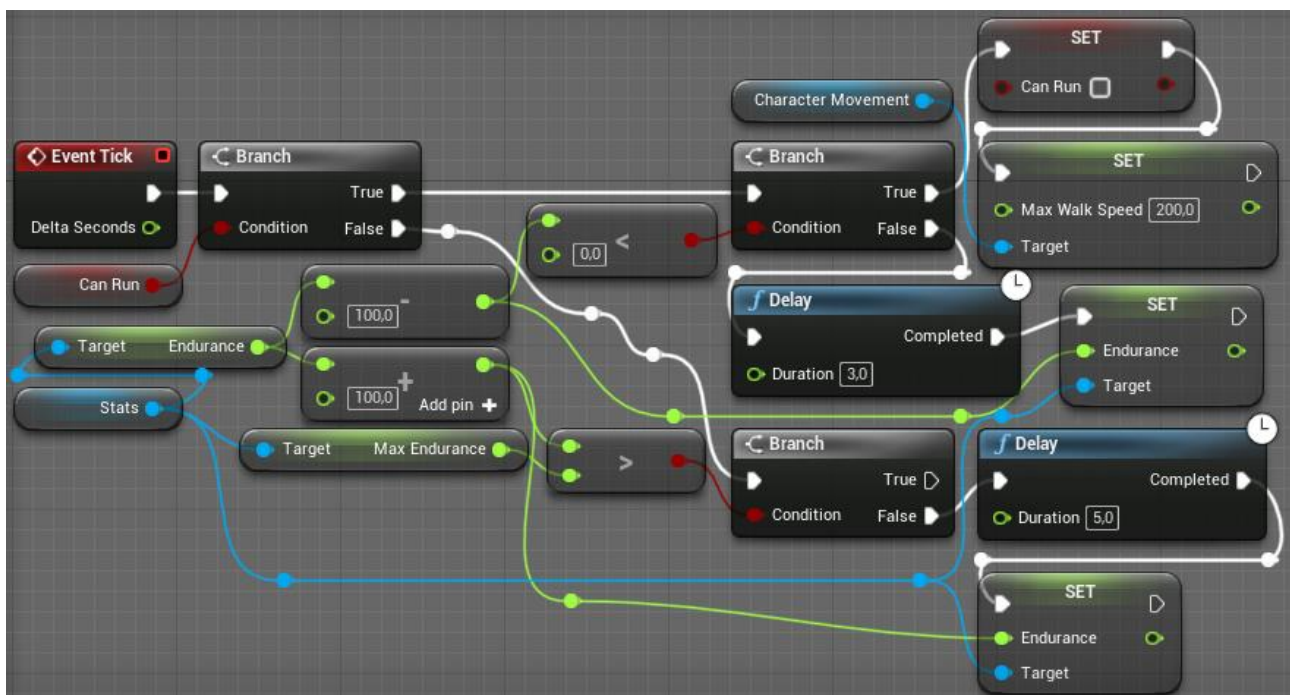


Рис. 11. Модуль логики для реализации игрового механизма «выносливость».

Принцип работы механизма «рассудок» в рассматриваемом приложении аналогичен механизму «здоровье» персонажа в подобных игровых программах. Была реализована следующая логика рассматриваемой опции: после столкновения персонажа с неигровыми персонажами (получение урона) значение рассудка снижается (рис. 12), в случае достижения шкалой «рассудок» отметки «0» – персонаж погибает (рис. 13).

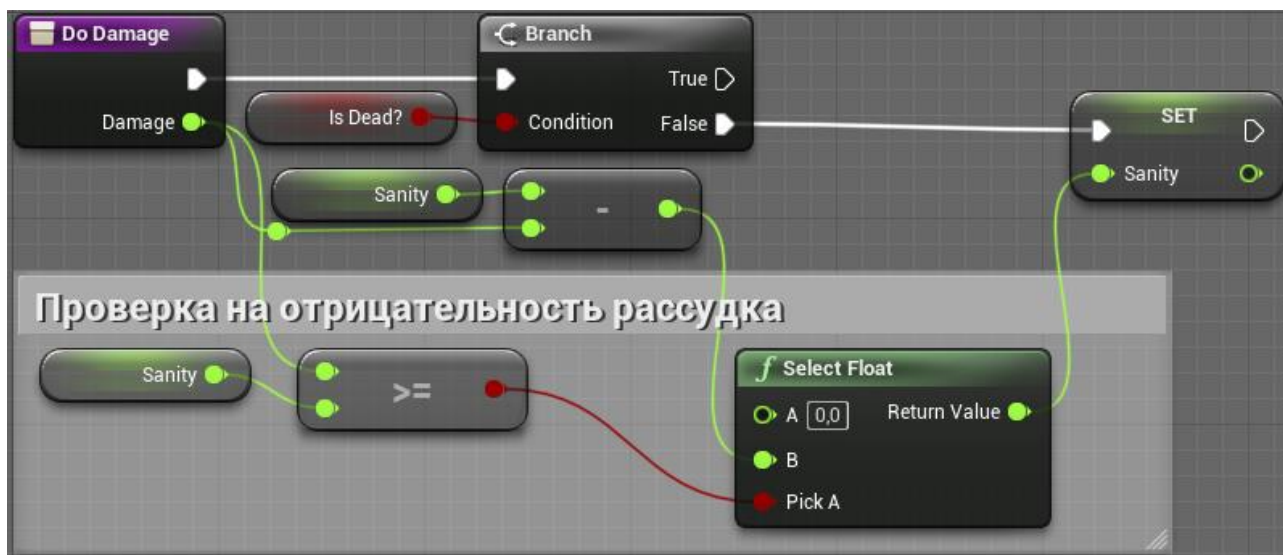


Рис. 12. Модуль логики для нанесения урона персонажу.

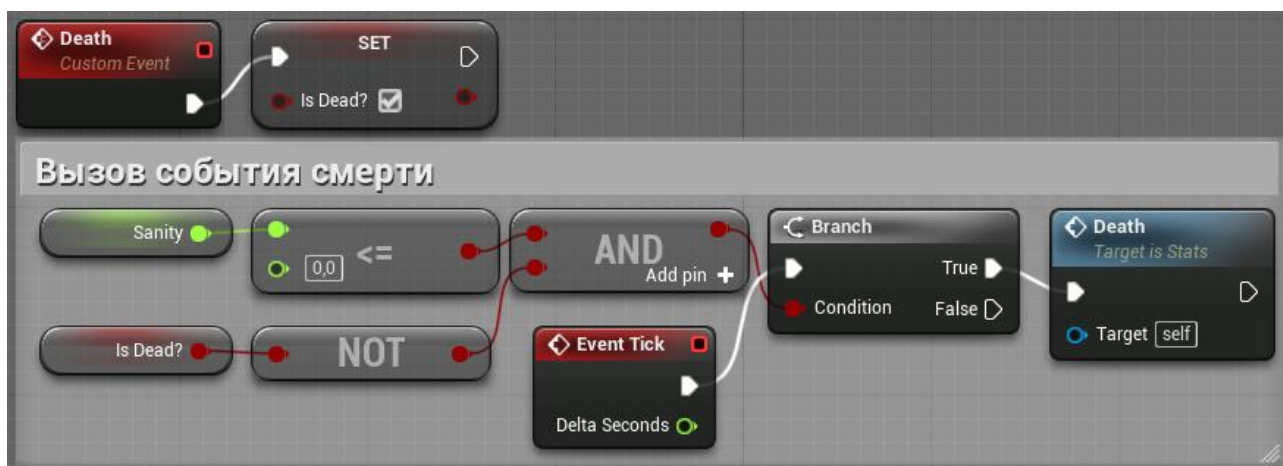


Рис. 13. Модуль логики вызова события смерти персонажа.

Мёртвый персонаж не может совершать каких-либо действий, поэтому авторами была создана функция для предотвращения таких ситуаций. Данная функция проверяет состояние персонажа и в случае его смерти блокирует любые его движения (рис. 14).



Рис. 14 Модуль логики блокировки любых действий персонажа в случае его смерти.

4) Разрушение объектов

Иногда пользователю необходимо будет найти определённые предметы в реализуемом приложении для дальнейшей работы в нём. Для усложнения поиска данные предметы будут спрятаны внутри других, например, в вазе. Чтобы достать их, персонаж должен разбить данную вазу или, другими словами, разрушить объект.

Чтобы при взаимодействии с определённым объектом главный герой мог его разрушить, первоначально потребовалось создать и настроить разрушаемый объект (рис. 15, красным прямоугольником обозначен триггер разрушения объекта, жёлтым прямоугольником – количество частей, на которые распадётся объект при столкновении). Далее авторами была реализована логика, позволяющая объекту, разрушиться при взаимодействии с персонажем (рис. 16).

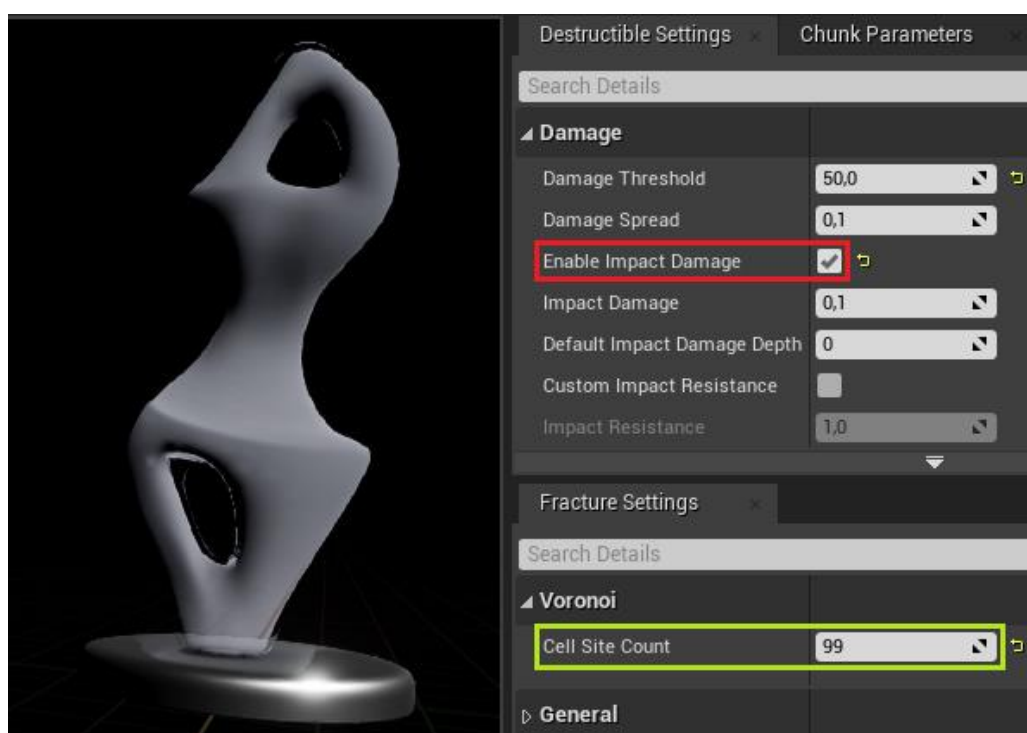


Рис. 15. Создание и настройка разрушаемого объекта.

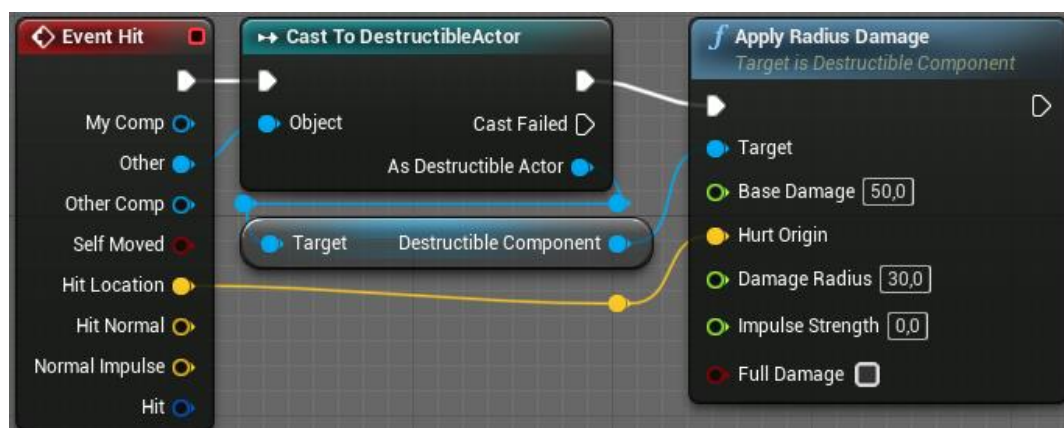


Рис. 16. Модуль логики разрушения объекта при столкновении с главным персонажем.

Заключение. Все задачи, поставленные в рамках данного исследования, были решены. Результатом работы является разработанное тестовое игровое приложение с описанным функционалом, позволившее в процессе своей реализации изучить основные принципы работы в игровом движке «Unreal Engine», а также особенности программирования на визуальном языке программирования «Blueprint».

СПИСОК ЛИТЕРАТУРЫ

1. Объемы игрового рынка в 2022 году могут превысить 200 млрд долларов [Электронный ресурс]. – Режим доступа: <https://club.dns-shop.ru/digest/obemyi-igrovogo-ryinka-v-2022-godu-mogut-prevyisit-200-mlrd-dolla/> (дата обращения: 02.11.2022).
2. Компьютерные и видеоигры (российский рынок) [Электронный ресурс]. – Режим доступа: [https://www.tadviser.ru/index.php/Статья:Компьютерные_и_видеоигры_\(российский_рынок\)](https://www.tadviser.ru/index.php/Статья:Компьютерные_и_видеоигры_(российский_рынок)) (дата обращения: 02.11.2022).
3. 5 лучших программ для создания собственной игры [Электронный ресурс]. – Режим доступа: <https://geeker.ru/games/programmy-dlya-sozdaniya-igr> (дата обращения: 02.11.2022).
4. Hardware and Software Specifications [Электронный ресурс]. – Режим доступа: <https://docs.unrealengine.com/4.27/RecommendedSpecifications/> (дата обращения: 02.11.2022).
5. Unreal Engine 4 [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/post/249965/> (дата обращения: 02.11.2022).
6. Основные приёмы Blueprint-программирования [Электронный ресурс]. – Режим доступа: <https://it-cube48.ru/основные-приёмы-blueprint-программирования/> (дата обращения: 02.11.2022).

ЗИНИНА С. Х., МЕДВЕДЕВА М. А.

**ДВУЦВЕТНЫЙ ГРАФ КАК СРЕДСТВО КЛАССИФИКАЦИИ
ДЕКАРТОВЫХ ПРОИЗВЕДЕНИЙ ГРУБЫХ ПРЕОБРАЗОВАНИЙ ОКРУЖНОСТИ¹**

Аннотация. В работе описан алгоритм сопоставления каждому диффеоморфизму, представляющему собой декартово произведение двух грубых преобразований окружности, двуцветного графа. Найдены необходимые и достаточные условия топологической сопряженности диффеоморфизмов, представляющих собой декартово произведение двух грубых преобразований окружности, посредством двуцветного графа.

Ключевые слова: грубое преобразование окружности, декартово произведение отображений, диффеоморфизм, топологическая сопряженность, двуцветный граф.

ZININA S. KH., MEDVEDEVA M. A.

**TWO-COLOR GRAPH AS MEANS OF CLASSIFICATION OF
CARTESIAN PRODUCTS OF ROUGH CIRCLE TRANSFORMATIONS**

Abstract. The paper describes an algorithm for matching each diffeomorphism, which is a Cartesian product of two rough transformations of a circle, a two-color graph. Necessary and sufficient conditions for topological conjugacy of diffeomorphisms of the class under consideration using a two-color graph are found.

Keywords: rough circle transformation, Cartesian product of maps, diffeomorphism, topological conjugacy, two-color graph.

Введение. Исследование грубых систем дифференциальных уравнений началось с работы [1] А. А. Андропова и Л. С. Понтрягина. В 1939 году А. Г. Майер в работе [2] ввел понятие грубости для динамических систем с дискретным временем (каскадов) на окружности и классифицировал грубые диффеоморфизмы окружности, это была одна из первых работ по топологической классификации динамических систем.

В 1959 году М. Пейшото в статье [3] обобщил результаты А. А. Андропова и Л. С. Понтрягина на произвольные ориентируемые замкнутые поверхности. В дальнейшем гиперболическая теория начала активно развиваться, С. Смейлом, Дж. Палисом, В. ди Мелу [4–6] была выстроена теория простейших структурно устойчивых систем, неблуждающее множество которых состоит из конечного числа орбит, называемых системами Морса-Смейла (в книге [7] дано систематизированное изложение классификации систем Морса-Смейла).

В 1971 году М. М. Пейшото в работе [8] формализовал понятие схемы Леонтович-

¹ Авторы благодарят О. В. Починку за консультации и внимание к работе.

Майера (введенное в 1955 г. в работе [9]) и доказал, что для систем Морса-Смейла на произвольных поверхностях, полным топологическим инвариантом является класс изоморфности его различающего графа. Результат Пейшото был обобщен В. З. Гринесом и А. Н. Безденежных для градиентно-подобных каскадов на ориентируемых поверхностях [10]. В работе [11] А. А. Ошемков и В. В. Шарко предложили поставить в соответствие потоку Морса-Смейла без замкнутых траекторий на поверхности трехцветный граф, который также является полным топологическим инвариантом, но по сравнению с графом Пейшото, описание и проверка изоморфности трехцветных графов является значительно более простой. В работе [12] были классифицированы градиентно-подобные диффеоморфизмы на поверхностях посредством трехцветного графа (T_f, P_f) . В статье [13] получена полная топологическая классификация n -кратных регулярных гомеоморфизмов окружности.

В настоящей работе рассматривается более узкий, по сравнению с работами [12; 13], класс G^* диффеоморфизмов, представляющих собой декартово произведение двух грубых преобразований окружности. В качестве претендента на роль топологического инварианта систем класса G^* предъясняется двуцветный граф и доказывается теорема о полноте множества найденных инвариантов.

Теорема. Диффеоморфизмы $f, f' \in G^*$ топологически сопряжены тогда и только тогда, когда двуцветные графы (D_f, P_f) и $(D_{f'}, P_{f'})$ изоморфны.

Пусть $f: M^2 \rightarrow M^2$ - диффеоморфизм класса G^* , представляющий собой декартово произведение двух грубых преобразований окружности.

Неблуждающее множество Ω диффеоморфизма $f: M^2 \rightarrow M^2$ представляется в следующем виде объединения множества стоковых (Ω_0), седловых (Ω_1) и источников (Ω_2) точек диффеоморфизма f , то есть: $\Omega = \Omega_0 \cup \Omega_1 \cup \Omega_2$. Множество $\Omega_1 \neq \emptyset$ в силу того, что диффеоморфизмы класса G^* являются декартовым произведением двух грубых преобразований окружности и имеют по крайней мере две седловые точки, фазовый портрет простейшего представителя класса G^* представлен на Рис. 3.

Следующее утверждение устанавливает топологический тип многообразий, допускающих диффеоморфизмы из класса G^* .

Утверждение. Пусть $f \in G^*$. Тогда M^2 гомеоморфно либо тору T^2 , либо бутылке Клейна K^2 .

Двуцветный граф.

Определение. Граф D называется *двуцветным графом*, если:

- 1) множество всех ребер графа D является объединением двух подмножеств, каждое из которых состоит из ребер одного цвета (цвета ребер обозначим буквами s, u);
- 2) каждая вершина графа D инцидентна в точности четырем ребрам, причем из них два

ребра цвета u , два ребра цвета s ;

3) граф не содержит петель.



Рис. 1. Примеры двуцветных графов с двумя, тремя и четырьмя вершинами.

На рисунке 1 приведены примеры простейших двуцветных графов; в силу определения двуцветного графа (он не содержит петель) минимально возможный двуцветный граф состоит из 2 вершин и 4 соединяющих их ребер.

Взаимно-однозначное отображение P графа D на себя, переводящее вершины в вершины с сохранением отношений инцидентности и цветности, называется *автоморфизмом графа D* . В дальнейшем мы будем понимать под символом (D, P) граф D , оснащенный автоморфизмом P .

Два двуцветных графа (D, P) и (D', P') назовем *изоморфными*, если существует взаимно-однозначное соответствие χ между множествами их вершин, сохраняющее отношения инцидентности и цветности, при этом сопрягающее автоморфизмы P и P' , то есть $P' \chi = \chi P$.

Опишем процесс построения двуцветного графа для диффеоморфизмов класса G^* , для этого проведем предварительные разбиения поверхности.

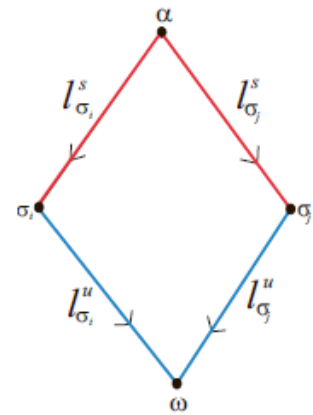


Рис. 2. Вид ячейки.

Положим $\tilde{M} = M^2 \setminus (W_{\Omega_0}^u \cup W_{\Omega_1}^u \cup W_{\Omega_1}^s \cup W_{\Omega_2}^s)$.

Множество $\tilde{M}$ представляется в виде объединения областей (ячеек), гомеоморфных открытому двумерному диску, граница каждой из которых имеет вид, изображенный на Рис. 2.

В границу каждой ячейки δ входят четыре периодические точки: источниковая точка α , седловые точки σ_i и σ_j , стоковая точка ω , а также устойчивые сепаратрисы $l_{\sigma_i}^s$ и $l_{\sigma_j}^s$ (s -кривые) с граничными точками α , σ_i и α , σ_j , неустойчивые сепаратрисы $l_{\sigma_i}^u$ и $l_{\sigma_j}^u$ (u -кривые) с граничными точками ω , σ_i и σ_j , ω (см. Рис. 2). Стороной ячейки назовем замыкание s - или u -кривой.

Периодом ячейки δ называется наименьшее натуральное число $k \in \mathbb{N}$, такое что $f^k(\delta) = \delta$.

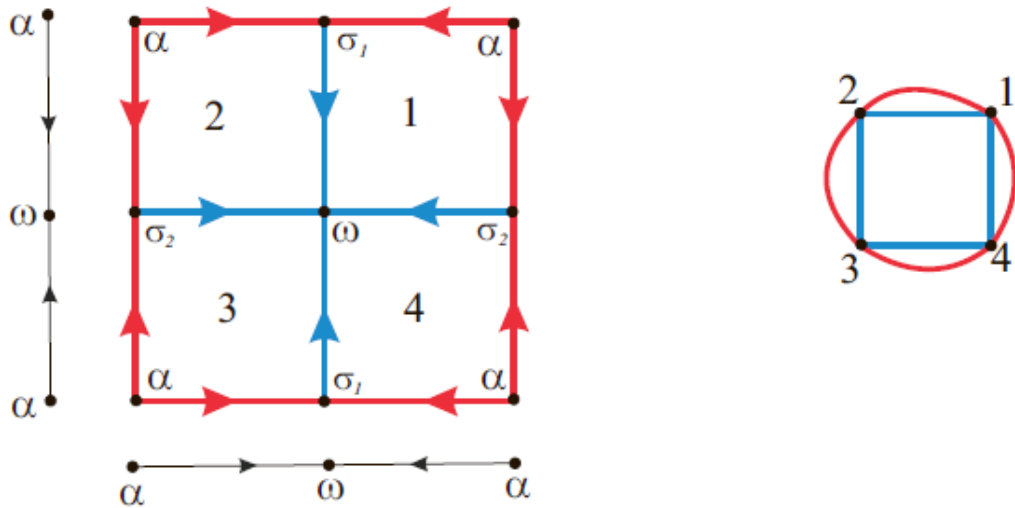


Рис. 3. Фазовый портрет диффеоморфизма класса G^* и соответствующий ему двуцветный граф.

Перейдем к непосредственному процессу построения двуцветного графа D_f , соответствующего диффеоморфизму $f \in G^*$ (см. Рис. 3-4).

- 1) Вершины графа D_f взаимно-однозначно соответствуют ячейкам множества Δ ;
- 2) две вершины графа инцидентны ребру цвета s или u , если соответствующие этим вершинам ячейки имеют общую s или u кривую.

Обозначим через B_f множество вершин графа D_f . Так как каждая ячейка имеет 4 стороны и из них две имеют цвет u , оставшиеся две - цвет s , то в вершине, соответствующей ячейке, сходятся 4 ребра, причем из них два ребра цвета u , два ребра цвета s .

Цикл, все ребра которого имеют цвет u (s) назовем u (s)-циклом; su -циклом назовем цикл, цвета всех ребер которого чередуются. Так как две вершины графа могут быть инцидентны двум ребрам различного цвета, то для однозначности условимся, что при описании su -цикла первое ребро имеет цвет s .

Поскольку все стороны ячейки различны, то есть любая сторона ячейки примыкает к стороне другой ячейки, то граф D_f не имеет петель.

Таким образом, граф D_f удовлетворяет определению двуцветного графа. Положим π_f взаимно-однозначное отображение множества ячеек диффеоморфизма f на множество вершин графа D_f . Диффеоморфизм f индуцирует на множестве вершин и ребер графа D_f автоморфизм $P_f = \pi_f f \pi_f^{-1}$.

Лемма. Двуцветный граф (D_f, P_f) обладает следующими свойствами:

- 1) граф D_f связан;
- 2) каждое ребро принадлежит единственному su -циклу длины 4; каждый s -цикл имеет

длину 4; каждый ω -цикл имеет длину 4.

3) автоморфизм P_f является периодическим с периодом m_f .

Для диффеоморфизмов класса G^* , представляющих собой декартово произведение двух грубых преобразований окружности, найдены необходимые и достаточные условия топологической сопряженности диффеоморфизмов посредством двуцветного графа, которые сформулированы в следующей теореме.

Теорема. Диффеоморфизмы f, f' из класса G^* топологически сопряжены тогда и только тогда, когда графы $(D_f; P_f), (D_{f'}; P_{f'})$ изоморфны.

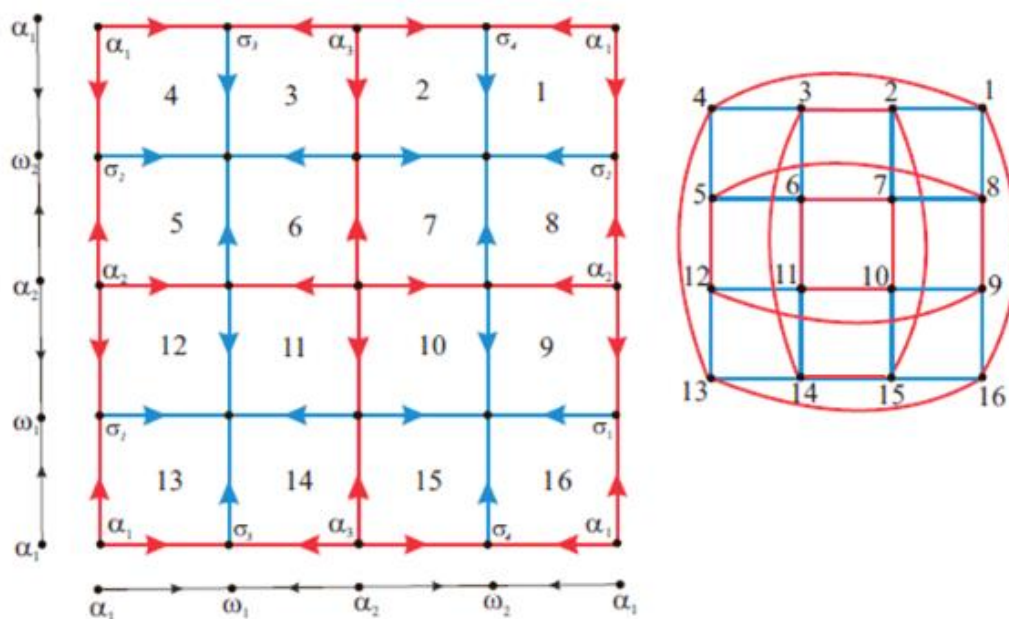


Рис. 4. Фазовый портрет диффеоморфизма класса G^*
и соответствующий ему двуцветный граф.

Заключение. В статье описан алгоритм сопоставления каждому диффеоморфизму, представляющему собой декартово произведение двух грубых преобразований окружности, двуцветного графа. Найдены необходимые и достаточные условия топологической сопряженности диффеоморфизмов, представляющих собой декартово произведение двух грубых преобразований окружности, посредством двуцветного графа.

СПИСОК ЛИТЕРАТУРЫ

1. Андронов А. А., Понтрягин Л. С. Грубые системы // Доклады академии наук СССР. – 1937. – Т. 14, № 5. – С. 247–250.

2. Майер А. Г. Грубое преобразование окружности в окружность // Ученые записки Горьк. гос. ун-та. – 1939. – Т. 12. – С. 215–229.
3. Peixoto M. M. On structural stability // Ann. Math. – 1959. – Vol. 69. – P. 199–222.
4. Смейл С. Дифференцируемые динамические системы // Успехи мат. наук. – 1970. – Т. 25. – С. 113–185.
5. Palis J. On Morse-Smale dynamical systems // Topology. – 1969. – Vol. 8, №4. – P. 385–404.
6. Palis J., Smale S. Structural stability theorems // Global Analysis, Proc. Sympos. Pure Math. – 1970. – Vol. 14. – P. 223–231.
7. Grines V., Medvedev T., Pochinka O. Dynamical Systems on 2- and 3-Manifolds. – Switzerland: Springer, 2016. – 313 p.
8. Peixoto M. M. On the classification of flows on two-manifolds // Dynamical systems. – 1973. – Acad. press N.Y. London. – P. 389–419.
9. Безденежных А. Н., Гринес В. З. Динамические свойства и топологическая классификация градиентноподобных диффеоморфизмов на двумерных многообразиях // Часть 1. Методы качественной теории дифференц. уравнений. Межвуз. темат. сб. научн. тр. под ред. Е.А. Лентович-Андроновой. – Горький, 1985. – С. 22–38.
10. Ошемков А. А., Шарко В. В. О классификации потоков Морса-Смейла на двумерных многообразиях // Математический сборник. – 1998. – Т. 189, № 8. – С. 93–140.
11. Гринес В. З., Капкаева С. Х., Починка О. В. Трехцветный граф как полный топологический инвариант для градиентно-подобных диффеоморфизмов поверхностей // Математический сборник. – 2014. – Т. 205, № 10. – С. 19–46.
12. Голикова И. В., Зинина С. Х. Топологическая сопряжённость n -кратных декартовых произведений грубых преобразований окружности // Известия высших учебных заведений. Прикладная нелинейная динамика. – 2021. – Т. 29, № 6. – С. 851–862.
13. Леонтович Е. А., Майер А. Г. О схеме, определяющей топологическую структуру разбиения на траектории // Доклады академии наук СССР. – 1955. – Т. 103, № 4. – С. 557–560.