

ПРИДНЕСТРОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМ. Т.Г. ШЕВЧЕНКО

**Рыбницкий филиал**

*Кафедра информатики и программной инженерии*

**НАУЧНЫЙ ПРОЕКТ**

**по дисциплине**

**«ПРОГРАММИРОВАНИЕ»**

**на тему:**

**«МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ДЛЯ ЗДОРОВОГО ОБРАЗА ЖИЗНИ»**

Студентов I курса направления  
«Программная инженерия»  
профиля «Разработка программно-  
информационных систем»

**Павлова Ивана Сергеевича**

**Стасюка Дениса Александрович**

**Синицина Александра**

**Александрович**

Научный руководитель:

старший преподаватель

**Гарбузняк Елена Сергеевна**

Рыбница, 2023

# АКТУАЛЬНОСТЬ, ОБЪЕКТ, ПРЕДМЕТ

**Актуальность** научного проекта определяется тем, что для ведения здорового образа жизни требуется учет калорий, потребляемых человеком в течении дня. В этом может помочь специальное приложение, предназначенная для подсчета количества калорий, съеденных человеком.

**Объект исследования** – здоровый образ жизни и бида его контроля

**Предмет исследования** – функциональные возможности мобильного приложения для здорового образа жизни

# ЗАДАЧИ

- изучить понятие здорового образа жизни и виды его контроля ;
- разработать архитектуру приложения и его функционал;
- реализовать мобильное приложение для платформы Android;
- провести тестирование и оптимизацию приложения.

# ИНСТРУМЕНТЫ РАЗРАБОТКИ

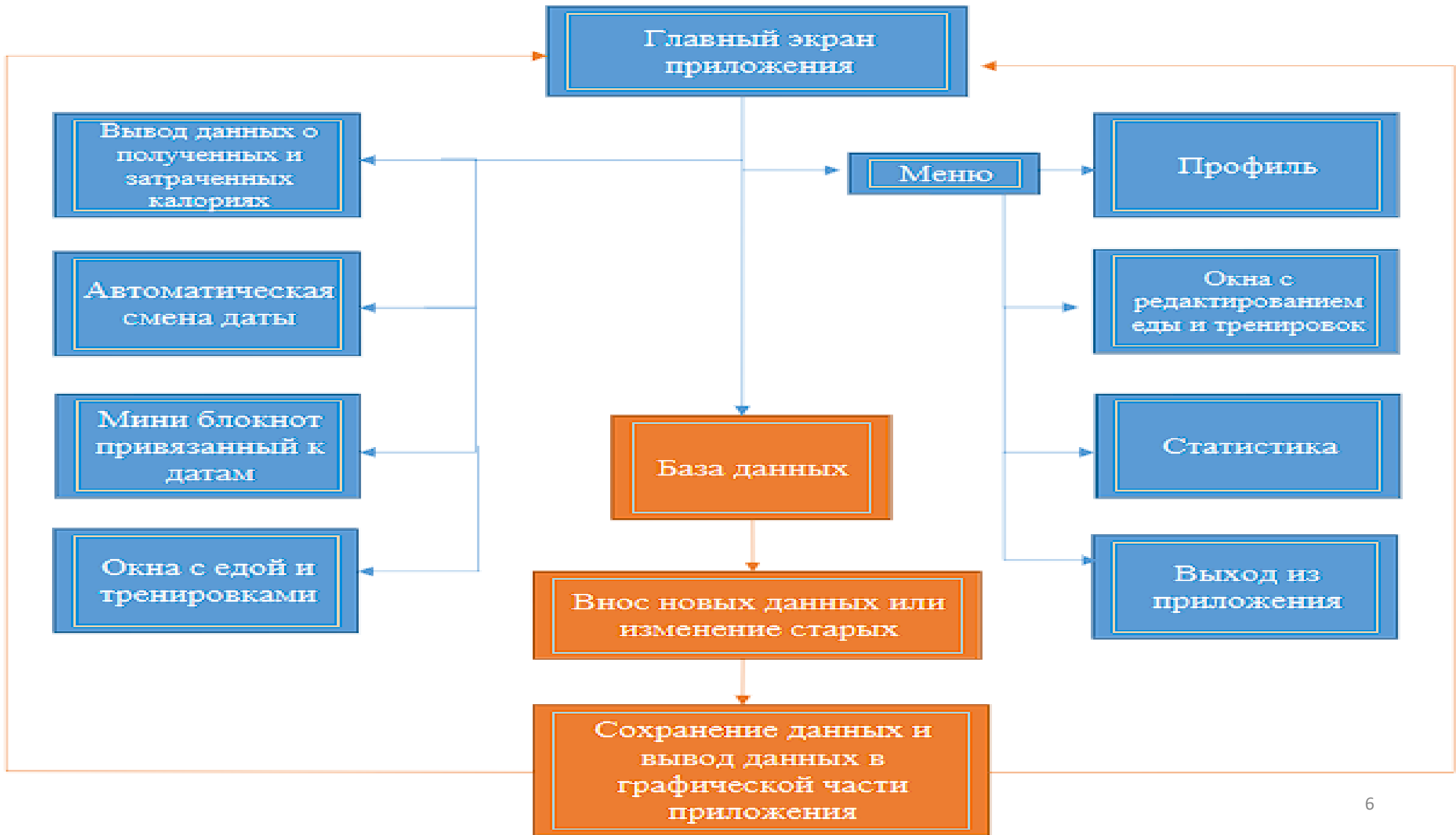
Язык Java



# СОЗДАНИЕ ПРОГРАММЫ

Основные этапы создания программы:

- создание меню программы;
- создание функций подсчета калорий и тренировок;
- создание базы данных для проекта.



## База данных

```
graph TD; DB[База данных] --- days; DB --- days_exercises; DB --- exercises; DB --- days_dishes; DB --- dishes;
```

### Таблица "days" (дни):

- ``_id`` (целое число) - идентификатор дня;
- ``date`` (текстовая строка) - дата;
- ``record`` (текстовая строка) - запись о дне.

### Таблица "days\_exercises" (упражнения дня):

- ``_id`` (целое число) - идентификатор записи упражнения дня;
- ``date`` (текстовая строка) - дата;
- ``exercise`` (текстовая строка) - название упражнения;
- ``time`` (целое число) - время упражнения;
- ``quantity`` (целое число) - количество упражнения.

### Таблица "exercises" (упражнения):

- ``_id`` (целое число) - идентификатор упражнения;
- ``exercise`` (текстовая строка) - название упражнения;
- ``quantity_coeff`` (целое число) - коэффициент количества упражнения;
- ``time_coeff`` (целое число) - коэффициент времени упражнения.

### Таблица "days\_dishes" (блюда дня):

- ``_id`` (целое число) - идентификатор записи блюда дня;
- ``date`` (текстовая строка) - дата;
- ``dish`` (текстовая строка) - название блюда;
- ``weight`` (целое число) - вес блюда.

### Таблица "dishes" (блюда):

- ``_id`` (целое число) - идентификатор блюда;
- ``dish`` (текстовая строка) - название блюда;
- ``calories`` (целое число) - количество калорий на 100 грамм блюда.

# ВНУТРЕННИЙ ФУНКЦИОНАЛ ПРОГРАММЫ

```
public class DataBase {
    1 usage
    private static final String DB_NAME = "CalorieCounterDB2";
    1 usage
    private static final int DB_VERSION = 1;

    8 usages
    private static final String DAYS_TABLE = "days";
    11 usages
    private static final String EXERCISE_TABLE = "exercises";
    10 usages
    private static final String DISH_TABLE = "dishes";
    9 usages
    private static final String DAYS_EXERCISE_TABLE = "days_exercises";
    9 usages
    private static final String DAYS_DISHES_TABLE = "days_dishes";

    2 usages
    public static final String COLUMN_ID = "_id";
    26 usages
    public static final String DAYS_COLUMN_DATE = "date";
    4 usages
    public static final String DAYS_COLUMN_RECORD = "record";
    26 usages
    public static final String EXERCISE_COLUMN_NAME = "exercise";
    7 usages
    public static final String EXERCISE_COLUMN_QUANTITY_COEFF = "quantity_coeff";
    7 usages
    public static final String EXERCISE_COLUMN_TIME_COEFF = "time_coeff";
```

```
2 usages
private static final String sampleExercisesNames[] = {"Pulling up the rear grip", "Push-ups", "Jumping rope", "Running on th
1 usage
private static final int sampleExercisesTimeCoeff[] = {1, 1, 0, 0, 4, 6};
1 usage
private static final int sampleExercisesQuantityCoeff[] = {12, 7, 4, 0, 0, 6};
2 usages
private static final String sampleDishesNames[] = {"Rice", "Army noodles", "Potato", "Hot dog", "Tea", "Cucumbers", "Lemon",
1 usage
private static final int sampleDishesCalories[] = {117, 248, 148, 44, 785, 154, 556, 232};
```



# ВНУТРЕННИЙ ФУНКЦИОНАЛ ПРОГРАММЫ

## Создание меню времени и автоматическое определение дня

```
public class Day {  
    2 usages  
    public static final SimpleDateFormat format = new SimpleDateFormat("dd.MM.yyyy");  
    1 usage  
    private static final String[] DAYS_OF_WEEK = {"Saturday", "Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday"};  
    1 usage  
    private static final String[] MONTH_OF_YEAR = {"January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December"};  
    2 usages  
    private final String dayOfWeek;  
    5 usages  
    private final MyDate date;  
    11 usages  
    private static Calendar calendar = Calendar.getInstance();  
    4 usages  
    private String record;  
    8 usages  
    private Map<Dish, Integer> dishes;  
    9 usages  
    private Map<Exercise, Pair<Integer, Integer>> exercises; //упражнение-время-количество
```

```
    public MyDate(Date date) {  
        super();  
        setTime(date.getTime());  
    }  
  
    @Override  
    public long getTime() {  
        setTime(super.getTime());  
        return super.getTime();  
    }  
  
    @Override  
    public void setTime(long milliseconds) {  
        Date date = new Date();  
        date.setTime(milliseconds);  
        Calendar calendar = Calendar.getInstance();  
        calendar.setTime(date);  
        calendar.set(Calendar.HOUR_OF_DAY, 1);  
        calendar.set(Calendar.MINUTE, 0);  
        calendar.set(Calendar.SECOND, 0);  
        calendar.set(Calendar.MILLISECOND, 0);  
        super.setTime(calendar.getTime().getTime());  
    }  
}
```

# ВНУТРЕННИЙ ФУНКЦИОНАЛ ПРОГРАММЫ

- код, который отображает список блюд из базы данных и позволяет выбирать элементы списка.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_dish);
    findViewById(R.id.from_dishes_to_menu).setOnClickListener(new ToWindowOnClickWithClosing(this, MyMenuActivity.class));

    db = DataBase.getDataBase(context: this); // open DB
    data=DataBase.cursorToArrayList(db.getDishes());

    // collation columns forming
    String[] from = new String[] {DataBase.DISH_COLUMN_NAME, DataBase.DISH_COLUMN_CALORIES_PER_100_GM }; // columns names
    int[] to = new int[] { R.id.db_item_name, R.id.db_item_right_text}; // places to write (View id)

    sAdapter = new SimpleAdapter(context: this, data, R.layout.database_item, from, to);
    listView= findViewById(R.id.dish_list_view);
    listView.setAdapter(sAdapter);
    listView.setChoiceMode(ListView.CHOICE_MODE_SINGLE);
    listView.setOnItemClickListener((parent, view, position, id) → {
        selectedElementId = id;
    });
}
```

# ВНУТРЕННИЙ ФУНКЦИОНАЛ ПРОГРАММЫ

## Создание профиля

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_profile);
    profile = Profile.getProfile(context: this);
    findViewById(R.id.from_profile_to_menu).setOnClickListener(new ToWindowOnClickWithClosing(this, MyMenuActivity.class));
    ((EditText) findViewById(R.id.profile_age)).setText(profile.getAge() + "");
    ((EditText) findViewById(R.id.profile_height)).setText(profile.getHeight() + "");
    ((EditText) findViewById(R.id.profile_weight)).setText(profile.getWeight() + "");
    ((RadioButton) findViewById(R.id.profile_woman_radio_button)).setChecked(!profile.getGender());
    ((EditText) findViewById(R.id.profile_aim_kal_number)).setText(profile.getAimCalorie() + "");
    optimalCalorieNumber = ((TextView) findViewById(R.id.profile_optimal_kal_number));
    optimalCalorieNumber.setText(profile.calculateCalories() + " " + getString(R.string.kilocalories));
}

public void onGenderRadioButtonClick(View view) {
```

# ТЕСТИРОВАНИЕ

## Профиль пользователя

Profile To Menu

Sex  Male  Female

Growth  cm

Weight  kg

Age  years

Daily calorie amount  
**1782 kcal**



Goal (calories per day)  
**1782kcal**

## Основное меню программы

Diary To Menu

Sunday  
10.06.2018

Got	Spent	Gap
440	5600	6942

Food Exercises

# Реализованные функции

- профиль пользователя с указанием данных о теле и автоматическом подсчёте калорий для поддержания веса;
- смена даты;
- база данных;
- интерфейс программы;
- блок тренировок и помощь по проекту.

# ЗАКЛЮЧЕНИЕ

- В ходе тестирования созданного программного продукта были выявлены следующие **недостатки**:
  - мало функций;
  - ошибки при работе с меню.
- При этом были отмечены такие преимущества, как:
  - простота интерфейса;

# ЗАКЛЮЧЕНИЕ

Представленное приложение для здорового образа жизни является ценным инструментом, способствующим улучшению качества жизни и повышению осведомленности о здоровье.

Оно предлагает пользователю широкий набор функций и ресурсов, которые могут помочь им принимать осознанные решения в отношении своего здоровья и благополучия.

В целом, разработанный программный продукт является неотъемлемым помощником в принятии здоровых и устойчивых решений, касающихся физического и психического благополучия.

Он может быть полезен для широкого круга пользователей, независимо от возраста и уровня физической подготовки, помогая им осознанно следить за своим здоровьем и принимать активное участие в своей жизни.