

Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и технологий
Высшая школа программной инженерии

ОТЧЕТ
Разработка приложения «Фитнес-трекер»
по дисциплине «Технологии разработки качественного ПО»

Выполнил
студент гр. 3530904/90321

Афанасов Е. П.

Руководитель

Смирнов Н. Г.

Санкт-Петербург

2022

Оглавление

Введение	3
Описание архитектуры приложения	4
Скриншоты приложения	6
Тестирование приложения	9
Использованные библиотеки и API	10
Вывод	11
Код программы	12

Введение

Цель работы – разработать приложение для подсчета сожжённых килокалорий во время физической активности.

Пользователь может указать вид тренировки. Доступны команды: начать тренировку (запускается таймер тренировки), закончить тренировку (таймер останавливается). За отработанное время высчитывается количество потраченных калорий по формуле $K \cdot t = \text{cal}$, где K - количество калорий в час, затрачиваемое на определенный вид тренировки, t - время, засеченное трекером.

От запуска к запуску программы данные должны сохраняться, а общее количество калорий суммироваться. Для этого реализовать персистенцию данных, поддержку нескольких профилей.

Описание архитектуры приложения

uml-диаграмма классов (создана в IntelliJ IDEA):

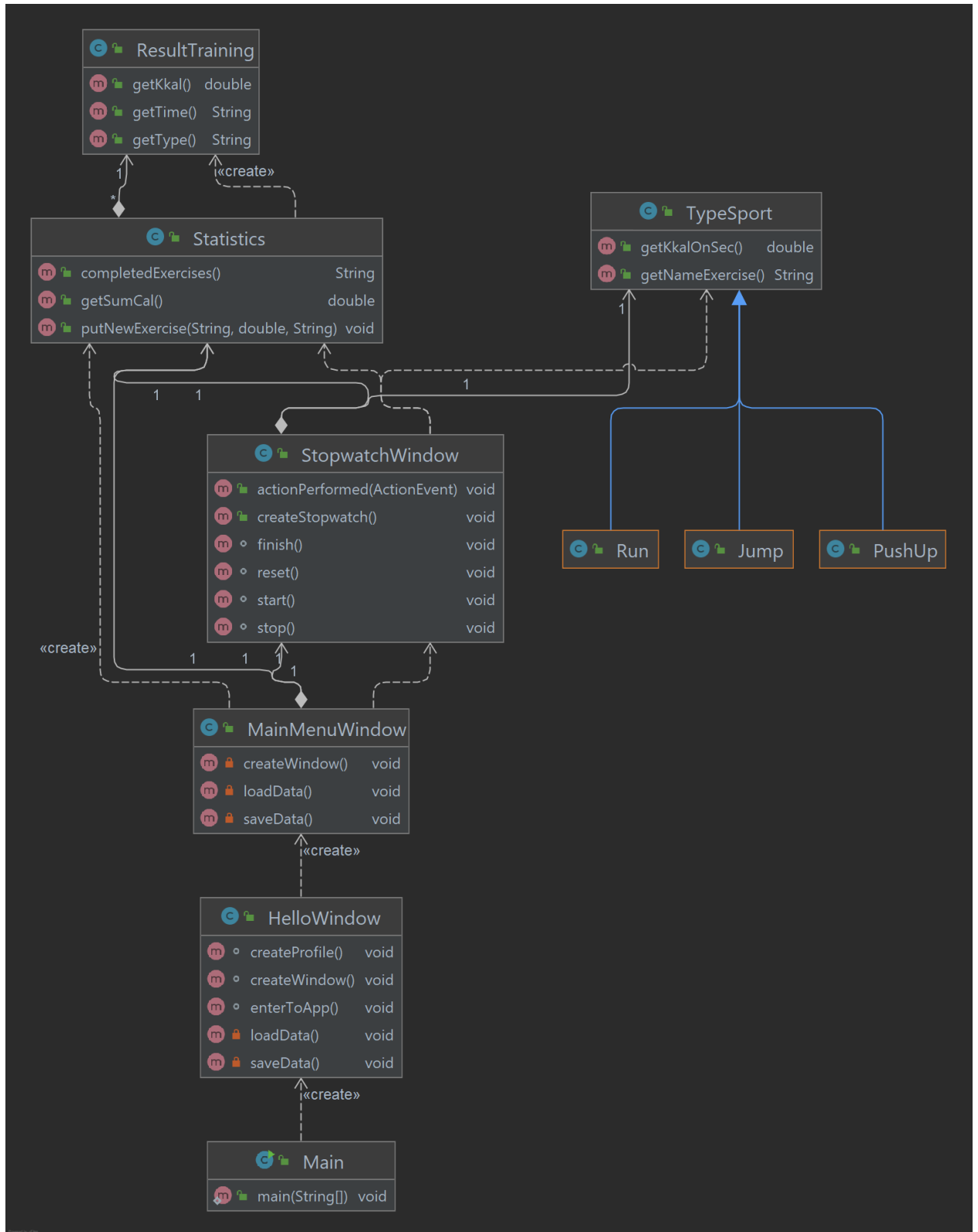
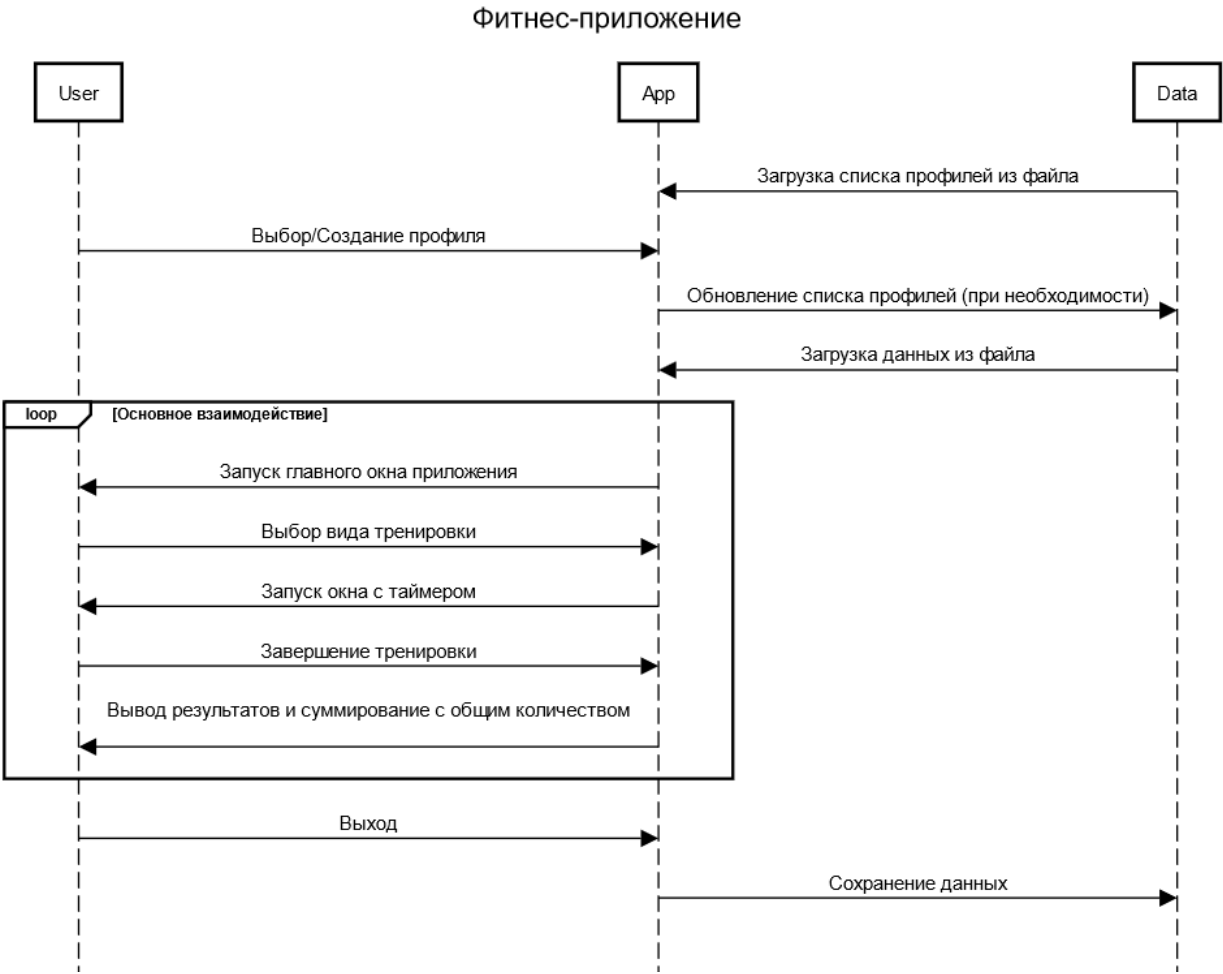
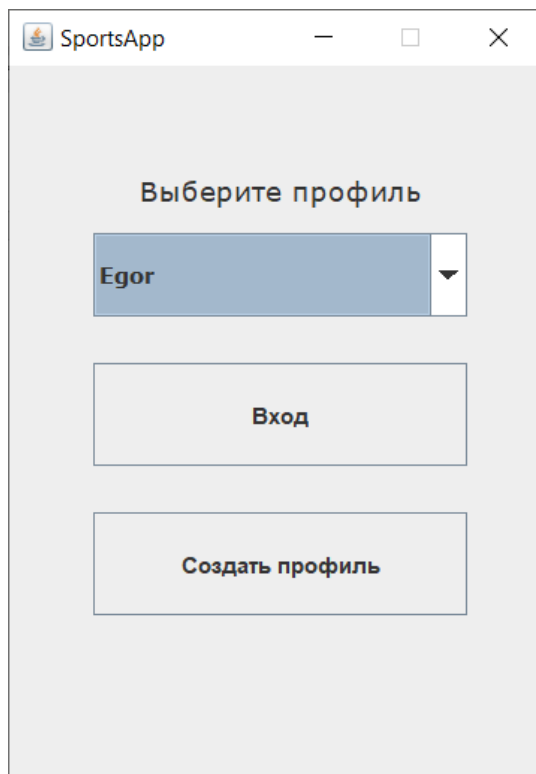


Диаграмма последовательности:

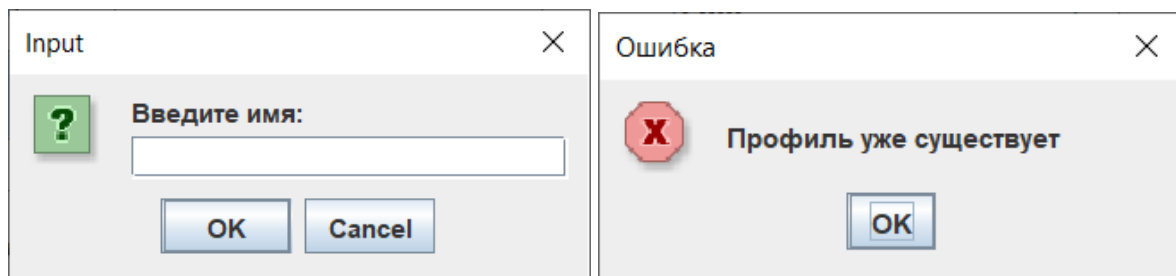


Скриншоты приложения

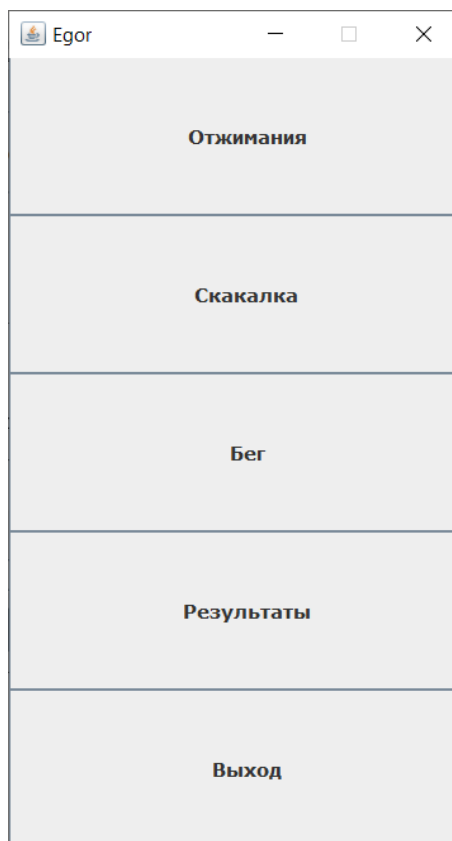
Окно выбора профиля



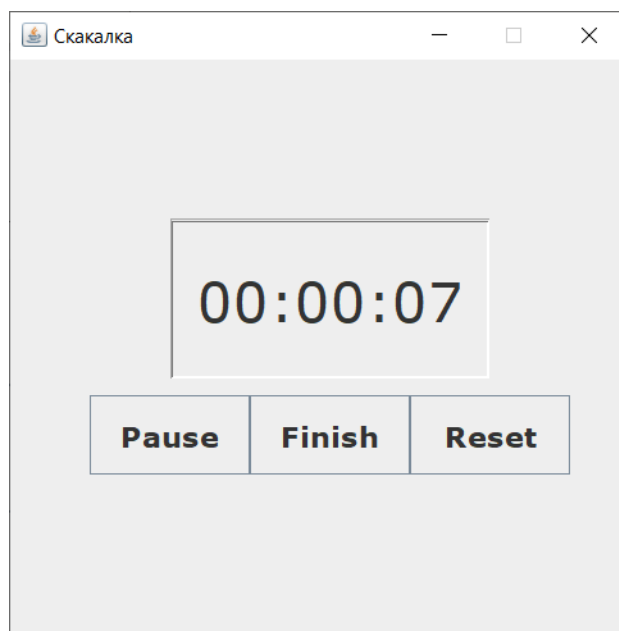
Создание профиля и ошибка



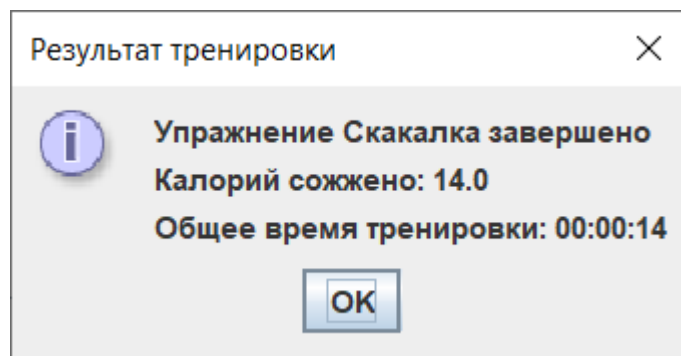
Основное окно приложения



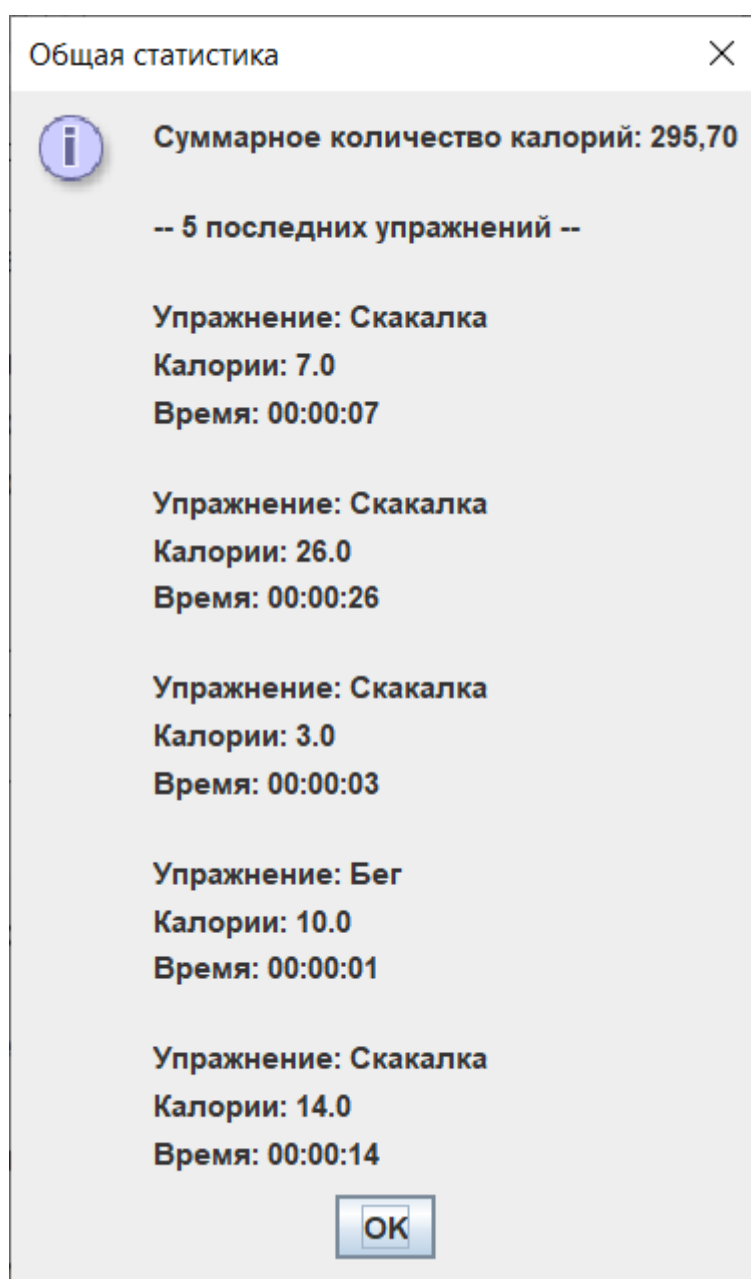
Окно с таймером



Сообщение о завершении тренировки



Вывод общей статистики

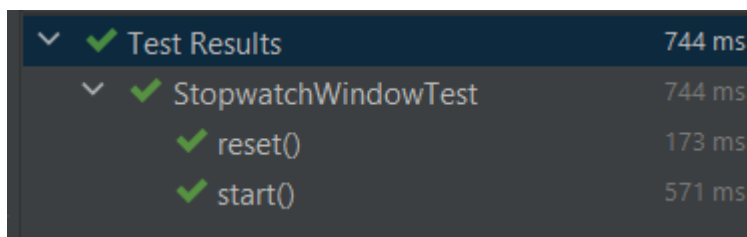


Тестирование приложения

В ходе разработки реализованы тесты для классов StopwatchWindow и Statistics.

StopWatchWindow – класс, отвечающий за вывод окна с таймером и его функционал.

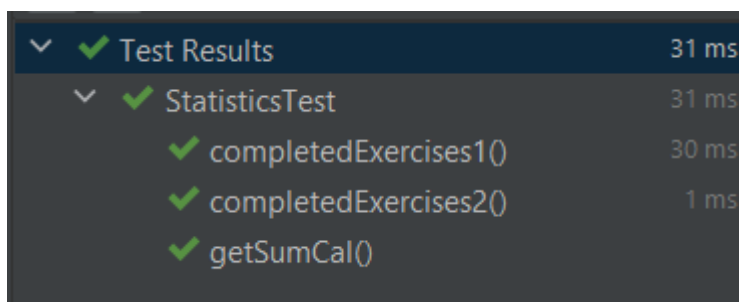
Протестированы функция сброса времени и изменение надписей на кнопке при нажатии на паузу/возобновление:



✓ Test Results	744 ms
✓ StopwatchWindowTest	744 ms
✓ reset()	173 ms
✓ start()	571 ms

Класс Statistics реализован для работы с массивом объектов класса ResultTraining, каждый такой объект содержит информацию о выполненном упражнении (тип, килокалории, время). В классе суммируется количество килокалорий за всё время и формируется список из последних 5 тренировок.

Протестированы метод, возвращающий список последних тренировок, и корректность вычисления суммы сожжённых килокалорий:



✓ Test Results	31 ms
✓ StatisticsTest	31 ms
✓ completedExercises1()	30 ms
✓ completedExercises2()	1 ms
✓ getSumCal()	

По скромным оценкам студента, покрытие кода тестами составляет 5–10%.

Использованные библиотеки и API

1. Java Serialization API – сохранение профилей и их данных.
2. Java Sound API – воспроизведение звукового эффекта в определенных условиях.
3. JUnit Jupiter API – тестирование приложения.
4. Swing – реализация графического интерфейса.

Вывод

При разработке приложения было реализовано 9 классов (и ещё 2 в рамках тестирования). Программа состоит из около 200 строк кода.

Достигнуты основные цели:

- Приложение считает количество калорий за отдельную тренировку и сумму за все время;
- Пользователь может выбрать вид тренировки;
- Данные после завершения приложения сохраняются;
- Существует система профилей.

Персистенция данных реализована при помощи ObjectOutputStream.

Графический интерфейс программы – библиотека Swing. Несмотря на простоту интерфейса, позволяет создать в кратчайшие сроки GUI со всем необходимым для приложения.

Код программы

- <https://github.com/Egorafnsv/sportApp>