

Обязательные для сдачи задания

Задание 1

Переработайте секундомер из домашнего задания 4 в объектно-ориентированном стиле. Для этого необходимо реализовать класс `Timer`, реализующий поведение секундомера.

Класс должен хранить следующие значения:

- таймер (объект, который создаётся методом `simplegui.create_timer()`);
- прошедшее от начала запуска время (в целом числе с разрешением 0.1 сек);
- полное число остановок и число остановок в ровную минуту (если реализуется тренажёр реакции).

Класс должен иметь следующие методы:

- конструктор – создаёт таймер и задаёт начальные нулевые значения для времени и счётчиков;
- метод для запуска таймера;
- метод для остановки таймера;
- метод сброса – обнуляет время и все счётчики;
- метод для форматирования времени – `format()`, который получает в качестве аргумента время в виде целого числа и возвращает отформатированную строку, согласно описания в домашнем задании 4.
- метод для рисования – получает в качестве аргумента холст и выводит на него все необходимые сообщения.

Для реализации методов рекомендуется использовать решение домашнего задания 4. Не забудьте, что у всех методов класса в качестве первого аргумента указывается `self`, и доступ к полям данных класса следует осуществлять через `self.имя_поля`.

Скорректируйте `doctest`'ы для метода `format` (или реализуйте их, если они не были реализованы в 4-ом задании). В `doctest`'е следует создать объект класса `Timer` и вызывать у него метод `format` для проверки разных значений, например, следующим образом:

```
>>> timer = Timer()
>>> timer.format(0)
'0:00.0'
>>> timer.format(11)
'0:01.1'
```

В основном тексте программы:

1. Создайте объект класса `Timer`.
2. Создайте окно приложения используя `simplegui.create_frame()`.
3. Создайте кнопки «Старт», «Стоп», «Сброс» и укажите для них в качестве обработчиков соответствующие методы класса `Timer` (для этого надо использовать синтаксис вида «имя_таймера.имя_метода»).
4. В качестве обработчика отрисовки холста назначьте метод рисования класса Таймер.
5. Запустите окно.

Обратите внимание, что в программе не должно остаться никаких глобальных переменных кроме таймера и окна.

Оценка обязательной части

1. Реализация секундомера с использованием класса.	60 %
2. Корректная работа функций «тренажера реакции» согласно домашнего задания 4.	10 %
3. Тесты для метода <code>format()</code> .	30%

Дополнительные задания

Задание 2

Добавьте в таймер возможность фиксации времени прохождения кругов. Для этого:

1. Добавьте новое поле данных – список, в который будут добавляться моменты прохождения круга.
2. Добавьте метод для фиксации времени круга – он должен добавить текущее значение времени в список (если секундомер активен). Время при этом не сбрасывается.
3. Создайте кнопку «Круг» указав разработанный метод в качестве обработчика.
4. Доработайте метод вывода на экран – он должен выводить зафиксированные в списке времена построчно, начиная от самого последнего к самому старому (первому в списке), используя метод `format()`.
5. Проверьте работу программы.
6. Реализуйте в классе `Timer` метод `format_laps()`, который получает на вход список со временами нажатия на кнопку «Круг» и выдаёт список строк, в каждой из которых указано: номер круга, время из входного списка и разница с предыдущим временем, например:

- входной список:

```
[10, 600, 700]
```

- выходной список:

```
['1 0:01.0 – 0:01.0',  
 '2 1:00.0 – 0:59.0',  
 '3 1:10.0 – 0:10.0']
```

Для форматирования времени в обоих случаях используйте метод `format()`.

Обязательно реализуйте `doctest`'ы для этого метода.

7. Используйте метод `format_laps()` для вывода кругов в методе вывода на экран.

Задание 3

Расширьте программу рисования квадратиками (дополнительное задание с прошлой недели) возможностью выводить окружности.

Выбор фигуры для рисования должен осуществляться нажатием соответствующих кнопок.

Для хранения информации о кружке реализуйте соответствующий класс. Рисование объектов осуществляйте с помощью специального метода рисования. Если Ваш класс квадратика не имел метода для рисования, добавьте в него такой метод.

Не забудьте, что при создании новой фигуры поверх старой, старая должна уничтожаться.

Оценка дополнительной части

Задание 2.1. Вывод времени начала круга – 20 %.

Задание 2.2. Реализация `format_laps` с тестами – 20 %.

Задание 3.1. Создание квадратиков и кружков – 30 %.

Задание 3.2. Рисование фигур методами классов – 15 %.

Задание 3.3. Удаление старых фигур при создании новых – 15 %.