1 Übungen

1.1 Pizza Pizza

Luigi betreibt einen kleinen Pizza-Lieferservice. Bisher bestand sein Geschäft darin, Kunden zu bedienen, die vorbeikommen oder anrufen.

In Zukunft möchte er seinen Kunden jedoch auch Online-Bestellungen anbieten. Um seine Organisation zu vereinfachen, beabsichtigt er jedoch, die Schnittstelle zur "Produktion" (Pizza-Bäckerei und -Lieferung) vorerst unverändert zu lassen: Bestellungen müssen in Papierform an die Küche geschickt werden.

Bestellungen, die künftig telefonisch oder im Geschäft eingehen, sollten ebenfalls mit dem neuen System eingegeben werden. Denn dann kann er sich immer und überall über sein Geschäft auf dem Laufenden halten: beim Friseur, am Strand oder in seinem Büro. Wofür ihm das neue System natürlich auch Zugang verschaffen muss.

Zum Bestellen ruft der Kunde im Webbrowser die Seite mit der Speisekarte des Pizza Dienstes auf. Dort trägt er bei den Pizzen, die er bestellen möchte, die Bestellmenge ein (Standard: 0). Anschließend gibt er Name, Anschrift, Telefonnummer und Email an.

Wenn er diese Daten "abschickt", wird eine Seite mit den zusammengefassten Bestallangaben inkl. Bestellnummer gezeigt. Erst wenn der Kunde diese Seite bestätigt, wird die Bestellung an den Pizza Dienst übermittelt und eine Quittungsemail an den Kunden geschickt.

Falls der Kunde unzufrieden ist mit der Zusammenfassung, kann er zurück zur Speisekarte gehen. Die ist dann mit seinen Angaben vorausgefüllt, so dass er nur noch korrigieren muss, was nicht gefällt. Für zukünftige Bestellungen werden seine persönlichen Angaben nach Bestätigung in einem Cookie vorgehalten.

Beim Pizza Dienst wird die Bestellung nach Empfang automatisch gedruckt.

1.2 Problem Triage

Entwickle eine Software, mit der eine Hotline Probleme an den Product Owner (PO) melden kann. Bei dieser Aufgabe soll nur die Problemanalyse mittels Softwarezelle durchgeführt werden.

Die Hotline wird von Programmanwendern im Problemfall konsultiert. Wenn die Hotline nicht mehr weiterhelfen kann, also aus ihrer Sicht ein Defizit in dem Programm vorliegt, meldet sie das als Probleme an den PO.
Pro Problem werden von der Hotline mindestens erfasst:

- Produkt – Wo ist das Problem aufgetreten?
- Beschreibung – Was sind die Symptome? Was sollte verbessert werden?
- Motivation des Anwenders – Warum sollte Abhilfe geschaffen werden?
- Hinweise zur Reproduktion
- Kunde
- Ansprechpartner des Kunden
- Kürzel des Hotline-Mitarbeiters
- Datum

Jeder Hotline-Mitarbeiter sieht seine gemeldeten Probleme. POs sehen alle gemeldeten Probleme.
Durch diese Probleme gehen die POs und beurteilen, wie mit ihnen weiter verfahren werden soll. Sie führen eine Triage durch und klassifizieren die Problemmeldungen:

- Rot: Unbrauchbares Problem, weil unverständlich formuliert
- Orange: Für das Problem gibt es schon eine Abhilfe (die die Hotline offensichtlich nicht kannte)
- Grün: Problem wird grundsätzlich akzeptiert und der weiteren Einplanung zugeführt
- Schwarz: Problem wird nicht weiter behandelt, z.B. weil es ökonomisch nicht sinnvoll ist

Die Übersicht der Probleme kann eingeschränkt werden nach Datum und/oder nach Klassen, z.B. nur rote+orangene der in einem Datumsbereich oder nur nicht klassifizierte Probleme. Bei der Klassifizierung wird im Problem vermerkt, wer wann klassifiziert hat. Außerdem kann der Klassifizierer eine Begründung angeben und insbesondere für grüne Probleme ein "Gewicht" (z.B. Anzahl profitierender Anwender, zu sparende Kosten oder gewonnener Umsatz).

Für einen Zeitraum kann eine Statistik generiert werden mit der Zahl der Probleme je Klasse pro Tag. Die Statistik wird in eine CSV-Datei folgender Form exportiert:

Datum	Rot	Orange	Grün	Schwarz
2.3.2014	3	5	1	4
3.3.2014	7	2	3	2

- Problemmeldungen können zur Weiterverarbeitung mit anderen Tools in eine CSV-Datei exportiert werden (Mehrzeilige Felder werden in Apostroph eingeschlossen).
- POs können jede Meldung löschen. Hotline-Mitarbeiter können nur noch **nicht** klassifizierte eigene Meldungen löschen.
- Die Anmeldung beim Softwaresystem kann informell geschehen.
- Die Unterscheidung zwischen PO und Hotline-Mitarbeiter soll während der Installation geschehen.

1.3 Die römischen Zahlen

Schreibe eine Konsolenanwendung, mit der römische Zahlen konvertiert werden können. Das folgende Beispiel zeigt einen möglichen Aufruf:

```
$ convertroman XLII
42
$ convertroman 2015
MMXV
```

Die zu konvertierende Zahl, entweder römisch oder arabisch, wird als Kommandozeilenparameter übergeben.
Das Ergebnis wird auf der Konsole ausgegeben.

Ist die römische Zahl ungültig, soll ein Fehlertext ausgegeben werden.

```
$ convertroman XLTII
Invalid roman digit found in "XLTII"
```

Wird der gültige Zahlenbereich von 1 bis 3000 überschritten, soll ebenso ein Fehlertext ausgegeben werden.

```
$ convertroman 15121
Invalid arabic number 15121; must be in range 1..3000
```

1.3.1 Erst Absprache mit dem Ausbilder und dann

Der Benutzer kann eine Textdatei als Parameter der Kommandozeile angeben, wie folgendes Beispiel zeigt:

```
$ convertroman -f numbers.txt
42
MMXV
```

Inhalt der Datei numbers.txt in diesem Beispiel:

```
XLII
2015
```

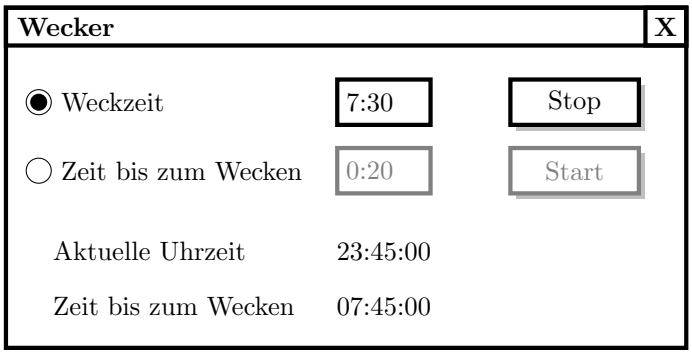
Tipp

Für leichtere Testbarkeit verwende Schnittstellen, um im Test die originale Implementierung auszutauschen (Fake).
Die Verwendung der Implementierungen aus dem Modul **Softwaredesign** sind erlaubt.

1.4 Der Wecker

Entwickle eine Anwendung, mit der man sich wecken lassen kann.
Es wird entweder eine Weckzeit eingegeben (Uhrzeit) oder eine Ruhezeit (Zeitraum bis zur Weckzeit).

Die Anwendung zeigt ständig die aktuelle Uhrzeit sekundengenau an. Sobald ein Weckauftrag durch Eingabe von Weckzeit bzw. Ruhezeit und Drücken eines Startknopfes gegeben wurde, zeigt sie auch die Restzeit bis zur Weckzeit sekundengenau.



Ist die Weckzeit erreicht, spielt die Anwendung eine Musikdatei mit Name weckton.wav (oder weckton.mp3) ab. Die Musikdatei wird abgespielt, bis der Wecker von Hand gestoppt wird. Natürlich kann der Wecker auch jederzeit vor Erreichen der Weckzeit gestoppt werden. Die Restzeit wird dann nicht länger angezeigt.

Tipp

Bestimme erst die Schnittstellen bevor die Implementierung begonnen wird. Eine grundlegendes Visual Studio Projekt wird durch den Ausbilder gestellt.