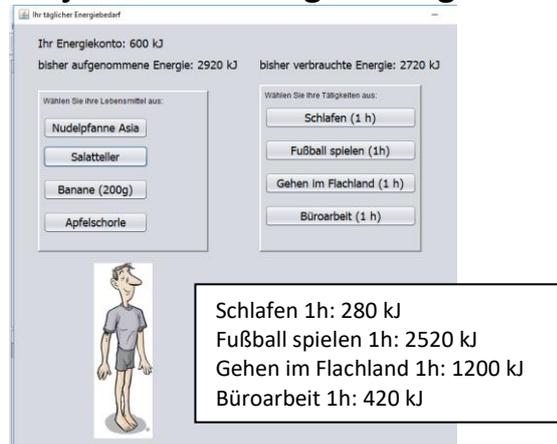
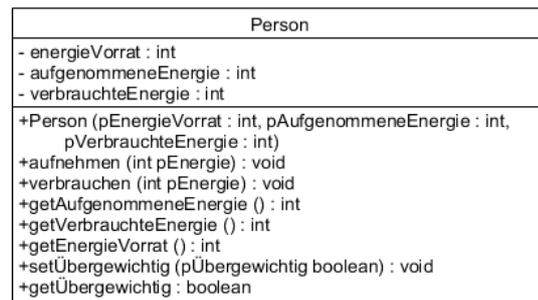


Projekt 3: Der Energiemanager



Der Energiemanager ist eine App, die für eine Person über die beim Essen und Trinken aufgenommene Energie und die bei verschiedene Tätigkeiten verbrauchte Energie Buch führen soll. Vereinfacht kann per Mausklick zwischen der Aufnahme von vier verschiedenen Lebensmitteln und dem Energieverbrauch bei vier verschiedenen Tätigkeiten ausgewählt werden. Nach jedem Mausklick soll die Energiereserve der Person aktualisiert werden. Beim Start des Programms besteht eine Energiereserve von 200 kJ durch Nahrungsmittelaufnahme vom vergangenen Tag.

Aufgabe 1: Direkt Rechts ist das **Implementationsdiagramm** der Klasse Person dargestellt. Im Implementationsdiagramm steht in der obersten Etage der Name der Klasse, in der zweiten Etage folgen alle Attribute, also die Eigenschaften der Person, die verwaltet werden müssen und in der untersten Etage stehen alle Methoden, die implementiert werden müssen. Anders als in Java stehen bei den Attributen die Datentypen getrennt von einem Doppelpunkt hinter den Namen der Attributen. Bei den Methoden steht bei solchen Methoden, die ein Ergebnis liefern sollen, der Datentyp ebenfalls hinter dem Methodennamen und der Parameterliste. Methoden, die kein Ergebnis zurückliefern sollen haben hier den Eintrag void.



Die App besteht aus 2 Klassen: Die Klasse **Gui** reagiert mit insgesamt acht Eventmethoden auf die Eingabe des Benutzers per Mausklick und steuert die Ausgabe, indem die Energiewerte jeweils aktualisiert werden. Die Klasse **Person** soll die drei Energiewerte der Person (Energievorrat, aufgenommene Energie, verbrauchte Energiealle) verwalten.

Importiere zunächst in Netbeans das Programm „**Energirechner**“, das du im Schülerverzeichnis findest. Die grafische Oberfläche der Klasse Gui ist bereits fertig, die Klasse Person ist jedoch noch leer.

- Ergänze die im **Klassendiagramm** der Klasse Person angegebenen Attribute und implementiere die aufgeführten Methoden.
- Implementiere in der Klasse Gui eine Methode energieDarstellen, die die Ausgabe der Energien auf dem Bildschirm regelt. Hier ist zum Beispiel folgender Befehl notwendig:
`energieKontoLab.setText ("Ihr Energiekonto: " + person1.getEnergieVorrat () + " kJ");`
- Erzeuge ein Objekt person1 der Klasse Person mit den oben angegeben Startwerten (200 kJ aufgenommene Energie, keine verbrauchte Energie, Einschätzung: kein Übergewicht). Ergänze die zehn vorhandenen Eventmethoden, hier ist z. B. bei der Asiapfanne der Befehl notwendig:
`person1.aufnehmen (2340); energieDarstellen();`
- Ist der Energievorrat größer als 3000 kJ und fühlt sich die Person übergewichtig, soll der Hinweis „Sie sollten dringend abnehmen!“ im tippLap erscheinen. **Tipp:** boolean ist ein VariablenTyp, der nur die Werte true oder false (richtig oder falsch) annehmen kann. `person1.setÜbergewichtig(true)` soll das Attribut übergewichtig auf true setzen. Entsprechend muss der Methodenaufruf in der Klasse Person so beginnen:
`public void setÜbergewichtig (boolean pÜbergewicht);`

Aufgabe 2: Für die folgende App sollen lediglich die **Implementationsdiagramme** von zwei sinnvoll gewählten Klassen aufgeschrieben werden, achte auf die formale Struktur eines Klassendiagramms, so wie sie in Aufgabe 1 beschrieben wurde: Ein Getränkeautomat kann Wasserflaschen (500 ml) zum Preis von 1,50€ und Flaschen mit Apfelschorle (200 ml) zum Preis von 1,80€ ausgeben. Der Automat ist zu Beginn mit 50 Wasserflaschen und 30 Apfelschorlen bestückt und enthält 20€ Wechselgeld. Mit einer App soll die Entnahme von Wasserflaschen und Apfelschorlen simuliert werden. Der Benutzer kann den Kauf mit zwei Buttons auslösen. Die App soll den Vorrat an beiden Getränkearten sowie die vorhandene Geldmenge aktualisieren.

Aufgabe 3: Denke dir einen eigenen Automaten aus, der mindestens vier Funktionen besitzt. Formuliere seine Aufgabe und gebe zwei sinnvoll gewählte Klassendiagramme an.

Zusatzaufgabe: Die Grafik soll sich verändern, wenn sich mehr als 3000 kJ Energievorrat angehäuft haben. Mit den Befehlen `bildLab.setIcon (dickerMann); bildLab.setIcon (dünnerMann);` kann man dies erreichen. Alternativ können auch die Bilder dickeFrau und dünneFrau verwendet werden.