

# Praktikum Objektorientierte Programmierung in C++ (WS 2023/2024)

[Dashboard](#) / [Meine Kurse](#) / [Wintersemester 2023/2024](#) / [Ingenieurwissenschaften](#)  
/ [Informatik und Angewandte Kognitionswissenschaften](#) / [Praktikum OOP in C++ WS 2023/2024](#) / [Aufgabe 6/Task 6](#)  
/ [A6: Aufgabe zur Abnahme in den Präsenz-Gruppen/Homework for Acceptance in the Presence Groups](#)

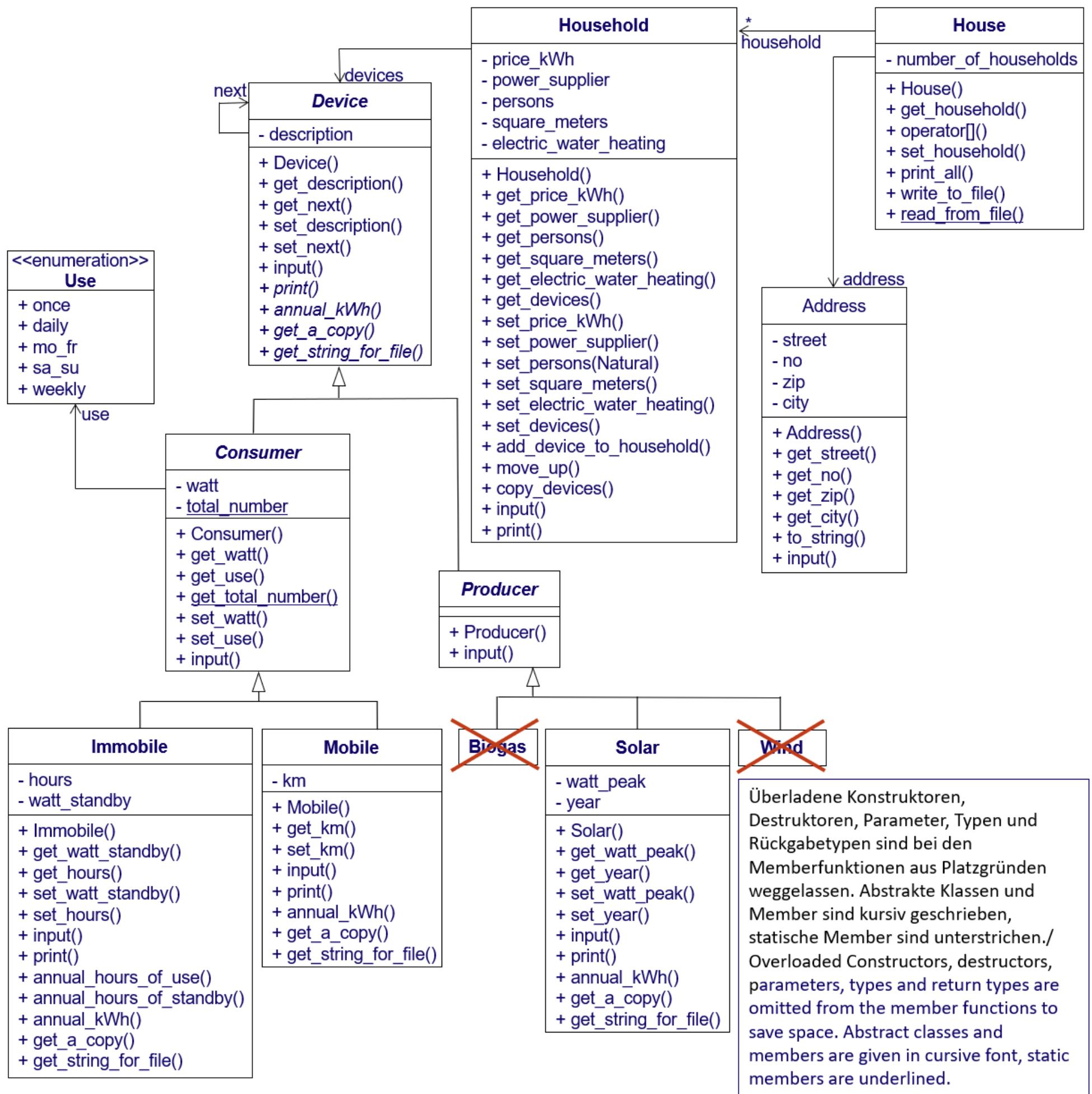
## A6: Aufgabe zur Abnahme in den Präsenz-Gruppen/Homework for Acceptance in the Presence Groups

In Aufgabe A5 wurden Klassen eingeführt für Stromverbraucher (**Consumer**), Haushalte (**Household**), Häuser (**House**) und deren Adressen (**Address**).

In Aufgabe A6 soll die Klassenstruktur umgebaut/erweitert werden auf eine Vererbungshierarchie für Geräte (**Device**) allgemein, zu denen Stromverbraucher (**Consumer**) und Stromerzeuger (**Producer**) gehören sollen. Bei den Stromverbrauchern werden mobile (**Mobile**) und nicht mobile (**Immobile**) unterschieden, als Beispiel für Stromerzeuger sollen Solaranlagen (**Solar**) hinzukommen. Das folgende UML-Klassenmodell zeigt die zu programmierenden Klassen mit ihren Membersn./

In task A5, classes were introduced for power consumers (**Consumer**), households (**Household**), houses (**House**) and their addresses (**Address**).

In task A6, the class structure is to be reorganised/extended to a single inheritance hierarchy for devices (**Device**) in general, which should include power consumers (**Consumer**) and power producers (**Producer**). A distinction is made between mobile (**Mobile**) and non-mobile (**Immobile**) power consumers, and solar systems (**Solar**) are to be added as an example of power producers. The following UML class model shows the classes to be programmed with their members.



- Definieren Sie eine abstrakte Klasse mit Namen *Device* (Gerät) mit folgenden Mitgliedern (verschieben Sie dabei geeignet aus der A5 Klasse *Consumer* die entsprechenden Member):/
 

Define an abstract class with name *Device* with the following members (move appropriate members from A5 class *Consumer*):

  - `private` Attribut namens *description* vom Typ C++-Zeichenkette./  
`private` attribute called *description* of type C++ string.
  - `private` Zeiger-Attribut namens *next* vom Typ *Device* zum Aufbau einer Liste von Geräten (Stromverbraucher und Stromerzeuger in einem Haushalt)./  
`private` pointer attribute named *next* of type *Device* to create a list of devices (power consumers and producers in a household).
  - `öffentlicher` überladener Konstruktor mit einem Zeiger vom Typ *Device* als Parameter und mit Defaultwert Nullzeiger, der das Zeiger-Attribut auf den Parameterwert initialisiert./  
`public` overloaded constructor with a pointer of type *Device* as a parameter with default value null pointer, which initialises the pointer attribute to the parameter value.
  - `öffentlicher virtueller` Destruktor, der eine Ausgabe wie in den Beispielen unten macht./  
`public virtual` destructor that produces an output as in the examples below.
  - `öffentliche` Member-Funktion mit Namen *get\_description* ohne Parameter, die den Wert des Zeichenketten-Attributs zurück liefert./  
`public` member function with name *get\_description* without parameter that returns the value of the string attribute.
  - `öffentliche` Member-Funktion mit Namen *get\_next* ohne Parameter, die den Wert des Zeiger-Attributs zurück liefert./  
`public` member function with name *get\_next* without parameter that returns the value of the pointer attribute.
  - `öffentliche` Member-Funktion mit Namen *set\_description* mit einer Zeichenkette als Parameter, die den Wert des Zeichenketten-Attributs auf den Parameterwert setzt./

public member function with name **set\_description** with a string as parameter, which sets the value of the string attribute to the parameter value.

- öffentliche Member-Funktion mit Namen **set\_next** mit einem Zeiger vom Typ **Device** als Parameter, die den Wert des Zeiger-Attributs auf den Parameterwert setzt./  
public member function with name **set\_next** with a pointer of type **Device** as parameter, which sets the value of the pointer attribute to the parameter value.
- virtuelle öffentliche Member-Funktion mit Namen **input** ohne Parameter und ohne Rückgabe, die eine Beschreibung für das Gerät einliest und im Attribut **description** speichert./  
public virtual member function with name **input** without parameter and without return, which reads in a description for the device and saves it in attribute **description**.
- öffentliche rein virtuelle, also dynamisch gebundene Member-Funktion mit Namen **print** mit einer ganzen Zahl und einer Gleitpunktzahl als Parameter ohne Rückgabe, die in den abgeleiteten Klassen implementiert werden muss./  
public pure virtual, i.e. dynamically bound member function with name **print** with an integer and a floating point number as parameters without return, which must be implemented in the derived classes.
- öffentliche rein virtuelle, also dynamisch gebundene Member-Funktion mit Namen **annual\_kWh** ohne Parameter und mit einer Gleitpunktzahl als Rückgabe, die in den abgeleiteten Klassen implementiert werden muss./  
public pure virtual, i.e. dynamically bound member function with name **annual\_kWh** without parameter and with a floating point number as return, which must be implemented in the derived classes.
- öffentliche rein virtuelle, also dynamisch gebundene Member-Funktion mit Namen **get\_a\_copy** ohne Parameter und mit einem Zeiger vom Typ **Device** als Rückgabe, die in den abgeleiteten Klassen implementiert werden muss./  
public pure virtual, i.e. dynamically bound member function with name **get\_a\_copy** without parameter and with a pointer of type **Device** as return, which must be implemented in the derived classes.
- öffentliche rein virtuelle, also dynamisch gebundene Member-Funktion mit Namen **get\_string\_for\_file** mit einem Zeichen als Parameter und einer C++-Zeichenkette als Rückgabe, die in den abgeleiteten Klassen implementiert werden muss./  
public pure virtual, i.e. dynamically bound member function named **get\_string\_for\_file** with one character as parameter and a C++ string as return, which must be implemented in the derived classes.

2. Ändern Sie die bisherige Klasse **Consumer** (Stromverbraucher) aus A5 um in eine abstrakte Klasse öffentlich abgeleitet von der Klasse **Device** mit:/

Change the previous class **Consumer** from A5 into an abstract class publicly derived from class **Device** with:

- behalten Sie die beiden privaten Attribute **watt** vom Typ Gleitpunktzahl und **use** vom Aufzählungstyp **Use**./  
keep the two private attributes **watt** of floating point number type and **use** of enumeration type **Use**.
- fügen Sie eine private statische ganzzahlige Klassenvariable **total\_number** hinzu, die mit dem Wert 0 initialisiert werden und die Anzahl der Objekte speichern soll./  
add a private static integer class variable **total\_number**, which shall be initialised with value 0 and store the number of objects.
- Definieren Sie einen öffentlichen überladenen Konstruktor mit einem Zeiger vom Typ **Device** als Parameter und mit Defaultwert Nullzeiger zur Initialisierung des Zeiger-Attributs der Oberklasse. Inkrementieren Sie im Rumpf die Klassenvariable, die das erzeugte Objekt mitzählen soll./  
Define a public overloaded constructor with a pointer of type **Device** as parameter with default value null pointer to initialise the pointer attribute of the superclass. In the body, increment the class variable counting the created objects.
- Definieren Sie einen öffentlichen virtuellen Destruktor, der eine Ausgabe wie in den Beispielen unten macht und die Klassenvariable dekrementiert./  
Define a public virtual destructor that produces an output as in the examples below and decrements the class variable.
- behalten Sie die öffentlichen Memberfunktionen **get\_watt**, **get\_use**, **set\_watt**, **set\_use** und fügen eine statische Klassenfunktion **get\_total\_number** ohne Parameter und mit einer ganzen Zahl als Rückgabe hinzu, die den Wert der Klassenvariable zurück liefert./  
keep the public member functions **get\_watt**, **get\_use**, **set\_watt**, **set\_use** and add a static class function **get\_total\_number** without parameter and with an integer as return, which returns the value of the class variable.
- virtuelle öffentliche Member-Funktion mit Namen **input** ohne Parameter, die zuerst die gleichnamige Member-Funktion der direkten Oberklasse aufruft und danach eine Wattzahl und die Häufigkeit der Benutzung einliest und in den Attributen **watt** und **use** speichert./  
virtual public member function with the name **input** without parameter, which first calls the member function of the same name of the direct superclass and then reads in a wattage and the frequency of use and saves both in the attributes **watt** and **use**.

3. Definieren Sie eine Klasse **Immobile** (nicht mobiler Stromverbraucher) öffentlich abgeleitet von der abstrakten Klasse **Consumer** mit:/  
Define a class **Immobile** (non-mobile power consumer) publicly derived from the abstract class **Consumer**:

- verschieben Sie die beiden privaten Attribute **watt\_standby** und **hours** vom Typ Gleitpunktzahl (aus der A5 Klasse **Consumer**) in diese Unterklasse./  
move the two private attributes **watt\_standby** and **hours** of type floating point number (from A5 class **Consumer**) to this subclass.
- definieren Sie einen öffentlichen überladenen Konstruktor mit einem Zeiger vom Typ **Device** als Parameter mit einem Defaultwert Nullzeiger zur Initialisierung des Zeiger-Attributs in der Oberklasse./  
define a public overloaded constructor with a pointer of type **Device** as parameter with a default value null pointer for initialising the pointer attribute in the superclass.



- definieren Sie einen virtuellen öffentlichen Destruktor, der eine Ausgabe wie in den Beispielen unten macht./  
define a virtual public destructor that produces an output as in the examples below.
  - übernehmen Sie Member-Funktionen `get_watt_standby`, `get_hours`, `set_watt_standby`, `set_hours`, `annual_hours_of_use`, `annual_hours_of_standby`, `annual_kWh`, `input` und `print` (aus der Klasse *Consumer* in A5). Passen Sie wo notwendig geeignet an, fügen Sie `print` als zweiten Parameter den Preis für eine Kilowattstunde hinzu, rufen Sie in `input` die gleichnamige Funktion der Oberklasse auf und erfragen danach die Werte der beiden hinzu gekommenen Attribute./  
Take member functions `get_watt_standby`, `get_hours`, `set_watt_standby`, `set_hours`, `annual_hours_of_use`, `annual_hours_of_standby`, `annual_kWh`, `input` and `print` (from the *Consumer* class in A5). Make suitable adjustments where necessary, add the price for a kilowatt hour as a second parameter to `print`, call in function `input` the function of the same name from the superclass and then query the values of the two added attributes.
  - definieren Sie eine virtuelle öffentliche Member-Funktion namens `get_a_copy` ohne Parameter und mit einem Zeiger vom Typ *Device* als Rückgabe. Erzeugen Sie im Rumpf ein neues Objekt der Klasse *Immobile* auf dem Heap, kopieren alle Attributwerte (Hinweis: möglich über eine einzige Anweisung `*kopie = *this;`) und geben die Adresse des Objekts als Funktionswert zurück (die Funktion wird weiter unten benötigt beim Kopieren aller Geräte in einem Haushalt in einen anderen Haushalt)./  
define a virtual public member function called `get_a_copy` without parameter and with a pointer of type *Device* as return. In the body, create a new object of the class *Immobile* on the heap, copy all attribute values (note: possible via a single `*copy = *this;` statement) and return the address of the object as function value (the function is required below when copying all devices in a household to another household).
  - definieren Sie eine virtuelle öffentliche Member-Funktion namens `get_string_for_file` mit einem Zeichen als Parameter und mit einer C++-Zeichenkette als Rückgabe. Konkatenieren Sie im Rumpf eine C++-Zeichenkette beginnend mit *Immobile* und hängen jeweils getrennt über das Zeichen im Parameter als Separatorzeichen die Werte aller Attribute (ausser von `next`) an die Zeichenkette an, so dass Sie (genutzt in der weiter unten definierten Funktion `write_to_file`) die komplette Zeile für dieses Objekt als eine Zeichenkette erhalten und als Funktionswert zurück geben (siehe Beispiele unten und Beispieldateien)./  
define a virtual public member function called `get_string_for_file` with a character as parameter and a C++ string as return. In the body, concatenate a C++ string starting with *Immobile* and append the values of all attributes (except `next`) to the character string separated by the separator character in the parameter, so that you receive a complete line for this object as character string (used in `write_to_file` function defined below) and return it as function value (see examples below and example files).
4. Definieren Sie eine Klasse *Mobile* (mobiler Stromverbraucher wie E-Auto, E-Bike, E-Scooter, ...) öffentlich abgeleitet von der abstrakten Klasse *Consumer* mit:/
- Define a class *Mobile* (mobile power consumer such as e-car, e-bike, e-scooter, ...) publicly derived from the abstract class *Consumer*:
- definieren Sie ein privates Attribut `km` vom Typ Gleitpunktzahl für die gefahrenen Kilometer./  
define a private attribute `km` of the floating point number type for the kilometres driven.
  - definieren Sie einen öffentlichen überladenen Konstruktor mit einem Zeiger vom Typ *Device* als Parameter mit Defaultwert Nullzeiger zur Initialisierung des Zeiger-Attributs in der Oberklasse./  
define a public overloaded constructor with a pointer of type *Device* as a parameter with default value null pointer to initialise the pointer attribute in the superclass.
  - definieren Sie einen virtuellen öffentlichen Destruktor, der eine Ausgabe wie in den Beispielen unten macht./  
define a virtual public destructor that produces an output as in the examples below.
  - definieren Sie zwei öffentlichen Member-Funktionen `get_km` und `set_km`, die den Wert des Attributs zurück liefern bzw. setzen sollen auf den Parameterwert, den `set_km` übergeben bekommt./  
define two public member functions `get_km` and `set_km`, which return the value of the attribute resp. set it to the parameter value passed to `set_km`.
  - definieren Sie eine öffentliche Member-Funktion namens `input` ohne Parameter und ohne Rückgabe. Rufen Sie im Rumpf die gleichnamige Funktion der Oberklasse auf und fragen danach einen Wert für das Attribut `km` ab./  
define a public member function called `input` without parameter and without return. In the body, call the superclass function of the same name and then query for a value for the attribute `km`.
  - definieren Sie eine virtuelle öffentliche Member-Funktion namens `annual_kWh` ohne Parameter und mit einer Gleitpunktzahl als Rückgabe für die im Jahr verbrauchten Kilowattstunden des mobilen Verbrauchers. Berechnen Sie als Rückgabewert im Rumpf ähnlich zu den nicht mobilen Verbrauchern den Jahres-Stromverbrauch über das entsprechend umgerechnete Produkt aus der Häufigkeit der Nutzung multipliziert mit den dabei jeweils gefahrenen Kilometern und mit den auf 100 km verbrauchten Kilowattstunden (siehe Beispiele unten)./  
Define a virtual public member function called `annual_kWh` without parameters and with a floating point number as return for the kilowatt hours consumed by the mobile consumer during the year. Calculate the annual power consumption as return value in the body similar to the non-mobile consumers, using the correspondingly converted product of the frequency of use multiplied by the kilometres driven and the kilowatt hours consumed per 100 km (see examples below).
  - definieren Sie eine virtuelle öffentliche Member-Funktion namens `print` ohne Rückgabe mit einer ganzen Zahl als erstem Parameter (Nummer in der Liste der Geräte) und einer Gleitpunktzahl als zweitem Parameter für den Preis einer Kilowattstunde. Geben Sie im Rumpf analog zu den nicht mobilen Verbrauchern die Nummer aus dem Parameter und die Beschreibung des mobilen Verbrauchers (Attribut `description` der Oberklasse *Device*) aus, die Adresse im Speicher, den Verbrauch an Kilowattstunden auf 100 km (Attribut `watt` der Oberklasse *Consumer* umgerechnet in Kilowatt), die Häufigkeit der Benutzung, die gefahrenen Kilometer, den Jahresverbrauch an Kilowattstunden und die Kosten dafür./  
define a virtual public member function called `print` without return with an integer as the first parameter (number in the list of

devices) and a floating point number as the second parameter for the price of a kilowatt hour. In the body in the same way as for non-mobile consumers, output the number from the parameter and the description of the mobile consumer (attribute **description** of the superclass **Device**), the address in the memory, the consumption of kilowatt hours per 100 km (attribute **watt** of the superclass **Consumer** converted to kilowatts), the frequency of use, the kilometres driven, the annual consumption of kilowatt hours and the costs for this.

- definieren Sie eine virtuelle öffentliche Member-Funktion namens **get\_a\_copy** ohne Parameter und mit einem Zeiger vom Typ **Device** als Rückgabe. Erzeugen Sie im Rumpf ein neues Objekt der Klasse **Mobile** auf dem Heap, kopieren alle Attributwerte (Hinweis: möglich über eine einzige Anweisung **\*kopie = \*this;**) und geben die Adresse des Objekts als Funktionswert zurück (die Funktion wird weiter unten benötigt beim Kopieren aller Geräte in einem Haushalt in einen anderen Haushalt)./  
define a virtual public member function called **get\_a\_copy** without parameter and with a pointer of type **Device** as return. In the body, create a new object of class **Mobile** on the heap, copy all attribute values (note: possible via a single statement **\*copy = \*this;**) and return the address of the object as function value (the function is required below when copying all devices in a household to another household).
  - definieren Sie eine virtuelle öffentliche Member-Funktion namens **get\_string\_for\_file** mit einem Zeichen als Parameter und mit einer C++-Zeichenkette als Rückgabe. Konkatenieren im Rumpf eine C++-Zeichenkette beginnend mit **Mobile** und hängen jeweils getrennt über das Zeichen im Parameter als Separatorzeichen die Werte aller Attribute (ausser von **next**) an die Zeichenkette an, so dass Sie (genutzt in der weiter unten definierten Funktion **write\_to\_file**) die komplette Zeile für dieses Objekt als eine Zeichenkette erhalten und als Funktionswert zurück geben (siehe Beispiele unten und Beispieldateien)./  
define a virtual public member function called **get\_string\_for\_file** with a character as parameter and a C++ string as return. In the body, concatenate a C++ character string starting with **Mobile** and append the values of all attributes (except **next**) to the character string separated by the separator character in the parameter, so that you receive a complete line for this object as character string (used in **write\_to\_file** function defined below) and return it as function value (see examples below and example files).
5. Definieren Sie eine abstrakte Klasse **Producer** (Stromerzeuger) öffentlich abgeleitet von der Klasse **Device** mit folgenden Mitgliedern:/  
Define an abstract class **Producer** (power generation) publicly derived from the class **Device** with the following members:
- öffentlicher überladener Konstruktor mit einem Zeiger vom Typ **Device** als Parameter mit Defaultwert Nullzeiger zur Initialisierung des Zeiger-Attributs in der Oberklasse./  
public overloaded constructor with a pointer of type **Device** as a parameter with default value null pointer to initialise the pointer attribute in the superclass.
  - virtueller öffentlicher Destruktor, der eine Ausgabe wie in den Beispielen unten macht./  
virtual public destructor that produces an output as in the examples below.
  - öffentliche Member-Funktion namens **input** ohne Parameter und ohne Rückgabe. Rufen Sie im Rumpf nur die gleichnamige Funktion der Oberklasse auf./  
public member function called **input** without parameter and without return. In the body, only call the superclass function of same name.
6. Definieren Sie eine Klasse **Solar** (Solaranlage) öffentlich abgeleitet von der abstrakten Klasse **Producer** mit folgenden Mitgliedern:/  
Define a class **Solar** (solar plant) publicly derived from the abstract class **Producer** with the following members:
- definieren Sie ein privates Attribut namens **watt\_peak** vom Typ Gleitpunktzahl für die installierte Leistung der Solaranlage./  
define a private attribute called **watt\_peak** of type floating point number for the installed power of the solar plant.
  - definieren Sie ein privates ganzzahliges Attribut namens **year** für das Jahr der Installation der Solaranlage./  
define a private integer attribute called **year** for the year in which the solar plant was installed.
  - definieren Sie einen öffentlichen überladenen Konstruktor mit einem Zeiger vom Typ **Device** als Parameter mit Defaultwert Nullzeiger zur Initialisierung des Zeiger-Attributs in der Oberklasse./  
define a public overloaded constructor with a pointer of type **Device** as parameter with default value null pointer to initialise the pointer attribute in the superclass.
  - definieren Sie einen virtuellen öffentlichen Destruktor, der eine Ausgabe wie in den Beispielen unten macht./  
define a virtual public destructor that produces an output as in the examples below.
  - definieren Sie vier öffentliche Member-Funktionen **get\_watt\_peak**, **get\_year**, **set\_watt\_peak** und **set\_year**, die die Werte der Attribute zurück liefern bzw. setzen sollen auf die Parameterwerte, die die Setter-Funktionen übergeben bekommen./  
define four public member functions **get\_watt\_peak**, **get\_year**, **set\_watt\_peak** and **set\_year**, which return resp. set the values of the attributes to the parameter value that are passed to the setter functions.
  - definieren Sie eine öffentliche Member-Funktion namens **input** ohne Parameter und ohne Rückgabe. Rufen Sie im Rumpf die gleichnamige Funktion der Oberklasse auf und fragen danach Werte für die Attribute **watt\_peak** und **year** ab (siehe Beispiele unten)./  
define a public member function called **input** without parameter and without return. Call the function of the same name of the superclass in the body and then query values for the attributes **watt\_peak** and **year** (see examples below).
  - definieren Sie eine virtuelle öffentliche Member-Funktion namens **annual\_kWh** ohne Parameter und mit einer Gleitpunktzahl als Rückgabe für die im Jahr erzeugten Kilowattstunden der Solaranlage. Als Vereinfachung kann für 1 Watt-Peak installierte Leistung 1 Watt berechnet werden. Berücksichtigen Sie bei der Berechnung zusätzlich die durch die Alterung der Solarzellen jedes Jahr um 0.5 % verminderte Leistung, als Beispiel weist also in diesem Jahr 2024 eine im Jahr 2014 installierte Solaranlage einen Verlust von 5 % auf. Geben Sie den Wert für die erzeugten Kilowattstunden mit einem negativen Vorzeichen versehen zurück, so dass bei der Summierung der Kilowattstunden aller Geräte diese erzeugten Kilowattstunden subtrahiert und nicht addiert werden (siehe Beispiele unten)./



define a virtual public member function called **annual\_kWh** without parameters and with a floating point number as return for the kilowatt hours generated by the solar plant in the year. As a simplification, 1 watt can be calculated for 1 watt peak of installed solar power. When making the calculation, also take into account a 0.5 % reduction in power output each year due to the ageing of the solar cells; for example, a solar system installed in 2014 will show a loss of 5 % in 2024. Return the value for the generated kilowatt hours with a negative sign so that when the kilowatt hours of all devices are summed up, these generated kilowatt hours are subtracted and not added (see examples below).

- definieren Sie eine virtuelle öffentliche Member-Funktion namens **print** ohne Rückgabe mit einer ganzen Zahl als erstem Parameter (Nummer in der Liste der Geräte) und einer Gleitpunktzahl als zweitem Parameter für den Preis einer Kilowattstunde. Geben Sie im Rumpf analog zu den Verbrauchern die Nummer aus dem Parameter und die Beschreibung des mobilen Verbrauchers (Attribut **description** der Oberklasse **Device**) aus, die Adresse im Speicher, die installierte Leistung, das Alter der Solaranlage, die jährliche Erzeugungsmenge und die gesparten Kosten (siehe Beispiele unten)./

define a virtual public member function called **print** without return with an integer as the first parameter (number in the list of devices) and a floating point number as second parameter for the price of a kilowatt hour. In the body, analogue to the consumers, print the number from the first parameter and the description of the mobile consumer (attribute **description** of the superclass **Device**), the address in the memory, the installed power, the age of the solar plant, the annual generation quantity and the costs saved (see examples below).

- definieren Sie eine virtuelle öffentliche Member-Funktion namens **get\_a\_copy** ohne Parameter und mit einem Zeiger vom Typ **Device** als Rückgabe. Erzeugen Sie im Rumpf ein neues Objekt der Klasse **Solar** auf dem Heap, kopieren alle Attributwerte (Hinweis: möglich über eine einzige Anweisung **\*kopie = \*this;**) und geben die Adresse des Objekts als Funktionswert zurück (die Funktion wird weiter unten benötigt beim Kopieren aller Geräte in einem Haushalt in einen anderen Haushalt)./

define a virtual public member function called **get\_a\_copy** without parameter and with a pointer of type **Device** as return. In the body, create a new object of class **Solar** on the heap, copy all attribute values (note: possible via a single **\*copy = \*this;** statement) and return the address of the object as function value (the function is required below when copying all devices in a household to another household).

- definieren Sie eine virtuelle öffentliche Member-Funktion namens **get\_string\_for\_file** mit einem Zeichen als Parameter und mit einer C++-Zeichenkette als Rückgabe. Konkatenieren im Rumpf eine C++-Zeichenkette beginnend mit **Solar** und hängen jeweils getrennt über das Zeichen im Parameter als Separatorzeichen die Werte aller Attribute (ausser von **next**) an die Zeichenkette an, so dass Sie (genutzt in der weiter unten definierten Funktion **write\_to\_file**) die komplette Zeile für dieses Objekt als eine Zeichenkette erhalten und als Funktionswert zurück geben (siehe Beispiele unten und Beispieldateien)./

define a virtual public member function called **get\_string\_for\_file** with a character as parameter and a C++ string as return. In the body, concatenate a C++ string starting with **Solar** and append the values of all attributes (except **next**) to the character string separated by the separator character in the parameter, so that you receive a complete line for this object as string (used in **write\_to\_file** function defined below) and return it as function value (see examples below and example files).

- Ändern Sie in der bisherigen Klasse **Household** (Haushalt) das Attribut **consumers** in ein Attribut namens **devices** vom Basis-Typ **Device** um und passen ebenfalls alle Member-Funktionen entsprechend an, also **get\_devices**, **set\_devices**, **add\_device\_to\_household**, **copy\_devices**, **move\_up**. Beachten Sie, dass im Unterschied zu **copy\_consumers** nun unterschiedliche Objekte in der Liste der Geräte vorkommen. Rufen Sie für eine Kopie der Liste von Geräten in der geänderten Funktion **copy\_devices** jeweils die virtuelle Funktion **get\_a\_copy** für Zeiger auf kopierte Objekte des gleichen Subtyps auf.  
In the previous class **Household**, change the attribute **consumers** to an attribute called **devices** of the base type **Device** and also adapt all member functions accordingly, i.e. **get\_devices**, **set\_devices**, **add\_device\_to\_household**, **copy\_devices**, **move\_up**. Note that, unlike **copy\_consumers**, different objects now appear in the list of devices. Therefore, copying a list of objects in a modified function **copy\_devices**, call the virtual function **get\_a\_copy** to receive pointers to copied objects of the same subtype.

- In der Klasse **House** müssen die beiden Funktionen zum Schreiben und Lesen von Dateien geändert/erweitert werden./

In the class **House**, the two functions for writing and reading files must be changed/extended.

- Ändern Sie in der Member-Funktion **write\_to\_file** die Versionsnummer am Anfang der Datei auf **A6**. Beim Durchlauf der jeweiligen Listen der Geräte in einem Haushalt können Sie nun einfacher als zuvor jeweils die von einer Nachricht **get\_string\_for\_file** an ein Gerät zurück gelieferte C++-Zeichenkette als Zeile in eine Datei schreiben./

In the member function **write\_to\_file**, change the version number at the beginning of the file to **A6**. More easily than before, when running through the respective list of devices in a household, you can write the C++ string returned by a message **get\_string\_for\_file** to a device as a line into the file.

- Ändern Sie die Funktion **read\_from\_file** um auf eine statische öffentliche Klassenfunktion, so dass Sie keine Nachricht mehr an ein Objekt senden müssen. Ändern Sie im Rumpf den Fall einer gelesenen Zeile beginnend mit **consumer** auf neu **Immobile** und beginnend mit **household** auf **Household**. Fügen Sie als weitere Fälle gelesene Zeilen beginnend mit **Mobile** und **Solar** hinzu./  
Change function **read\_from\_file** to a static public class function so that you no longer need to send a message to an object. In the body, change the case for reading a line starting with **consumer** to new **Immobile** and starting with **household** to new **Household**. Add further cases for reading lines starting with **Mobile** and **Solar**.

- Ändern Sie die Funktion **main** folgendermassen ab:/

Modify function **main** as follows:

- Ändern/Ergänzen Sie Menüpunkte für die Eingabe von Geräten:/  
Change/add menu items for inputting devices:  
**i input immobile consumer** (bisher **input consumer**/former **input consumer**)

```
m input mobile consumer
```

```
s input solar producer
```

- Rufen Sie im Menüpunkt/Directly call in menu entry

```
r read data from file
```

direkt die entsprechende Klassen-Funktion auf und weisen den von dieser zurück gelieferten Zeiger der Variable für das Haus zu./  
the corresponding class function and assign the pointer returned by it to the variable for the house.

### Hinweis

Überlegen Sie, wie Sie Ihr eigenes Programm am besten und geschicktesten schrittweise ändern und erweitern und testen können.

Beispielsweise könnten Sie zuerst `read_from_file` in eine Klassenfunktion umwandeln, in `main` entsprechend ändern und Ihr Programm austesten.

Danach könnten Sie überlegen, erstmal nur die Klassen *Device*, *Consumer* und *Immobile* auf Basis der vorherigen Version A5 zu programmieren und wieder Ihr Programm nach Änderungen in `main` auszutesten.

Auf dieser Grundlage können Sie dann analog die neuen Klassen *Producer*, *Mobile* und *Solar* nach und nach hinzufügen, danach jeweils die Funktionen `write_to_file` und `read_from_file` erweitern und so zum fertigen Programm kommen./

### Note

Think about how you can best and most skilfully change and extend and test your own program step by step.

For example, you could first convert `read_from_file` into a class function, change in `main` accordingly and test your program.

Then you could consider programming only the classes *Device*, *Consumer* and *Immobile* on the basis of the previous version A5 and testing your program after modifying `main` again.

On this basis, you can then add the new classes *Producer*, *Mobile* and *Solar* one by one, afterwards extend functions `write_to_file` and `read_from_file` and thus arrive at the finished program.

### Beispiel Programmlauf 1/Example Program Run 1

CALCULATION OF AVERAGE POWER COSTS FOR A HOUSE – CLASS VERSION WITH INHERITANCE

```
q quit
d delete house
h house initialisation
i input immobile consumer
m input mobile consumer
s input solar producer
u move up device
p print household
a print all households
n new household
c copy all devices (added to already existing ones)
r read data from file
w write data into file
>> h
how many households does the house have? 1
what is the street name? Teststrasse
what is house number? 123
what is zip code? 45678
what is the city name? Testort
Address Teststrasse 123, 45678 Testort at address 0x62fd20 deleted
Address Teststrasse 123, 45678 Testort at address 0x62fb70 deleted
q quit
d delete house
h house initialisation
i input immobile consumer
m input mobile consumer
s input solar producer
u move up device
p print household
a print all households
n new household
c copy all devices (added to already existing ones)
r read data from file
w write data into file
>> a
=====
                        H O U S E
=====
                        (this: 0x6c1a00)
                        address: Teststrasse 123, 45678 Testort
                        number of households: 1
                        total number of consumers: 0
=====
q quit
d delete house
h house initialisation
i input immobile consumer
m input mobile consumer
s input solar producer
u move up device
p print household
a print all households
n new household
c copy all devices (added to already existing ones)
r read data from file
w write data into file
>> n
number of household? 0
how many square metres does the household have? 100
how many persons live in this household? 1
is hot water heated using electricity? (y(es) or n(o)) y
what is the price for one kWh in EUR? 0.4
who is the power supplier? RWE
q quit
d delete house
h house initialisation
i input immobile consumer
m input mobile consumer
s input solar producer
u move up device
p print household
a print all households
n new household
c copy all devices (added to already existing ones)
r read data from file
```



```
w write data into file
>> a

=====
                        H O U S E
=====

                        (this: 0x6c1a00)
                        address: Teststrasse 123, 45678 Testort
                        number of households: 1
                        total number of consumers: 0
H O U S E H O L D   N O   0   P O W E R   C O N S U M P T I O N
=====

                        (this: 0x6c17a0)
                        price for one kWh: 40.00 ct/kWh
                        power supplier: RWE
                        square metres: 100.00 qm
                        persons: 1
                        water heated using electricity: yes
                        list of devices

=====

                        power consumption square meters: 900.0 kWh
                        power consumption all persons: 550.0 kWh
                        total annual power consumption: 1450.0 kWh
                        total annual power costs: 580.00 EUR
=====

q quit
d delete house
h house initialisation
i input immobile consumer
m input mobile consumer
s input solar producer
u move up device
p print household
a print all households
n new household
c copy all devices (added to already existing ones)
r read data from file
w write data into file
>> s
number of household? 0
what is the description of the power consumer? Balcony Power Plant
how many watt peak have been installed? 800
in which year the solar modules are installed? 2014
q quit
d delete house
h house initialisation
i input immobile consumer
m input mobile consumer
s input solar producer
u move up device
p print household
a print all households
n new household
c copy all devices (added to already existing ones)
r read data from file
w write data into file
>> a

=====
                        H O U S E
=====

                        (this: 0x6c1a00)
                        address: Teststrasse 123, 45678 Testort
                        number of households: 1
                        total number of consumers: 0
H O U S E H O L D   N O   0   P O W E R   C O N S U M P T I O N
=====

                        (this: 0x6c17a0)
                        price for one kWh: 40.00 ct/kWh
                        power supplier: RWE
                        square metres: 100.00 qm
                        persons: 1
                        water heated using electricity: yes
                        list of devices

=====

                        1: Balcony Power Plant
                        (this: 0x6c5ce0)
```

```
solar cells installed power: 800.00 watt_peak
age of solar cells: 10 years
annual production: -760.0 kWh
annual savings: -304.00 EUR
-----
power consumption square meters: 900.0 kWh
power consumption all persons: 550.0 kWh
total annual power consumption: 690.0 kWh
total annual power costs: 276.00 EUR
=====
q quit
d delete house
h house initialisation
i input immobile consumer
m input mobile consumer
s input solar producer
u move up device
p print household
a print all households
n new household
c copy all devices (added to already existing ones)
r read data from file
w write data into file
>> w
input file name: house1.csv
input separator character: ;
output file "house1.csv" opened...
output file "house1.csv" closed
q quit
d delete house
h house initialisation
i input immobile consumer
m input mobile consumer
s input solar producer
u move up device
p print household
a print all households
n new household
c copy all devices (added to already existing ones)
r read data from file
w write data into file
>> m
number of household? 0
what is the description of the power consumer? E-SUV
how many watt it will have? 21500
how often it will be used?
daily (d)
mo_fr (m)
once (o)
sa_su (s)
weekly (w)? m
how many km will be driven? 100
q quit
d delete house
h house initialisation
i input immobile consumer
m input mobile consumer
s input solar producer
u move up device
p print household
a print all households
n new household
c copy all devices (added to already existing ones)
r read data from file
w write data into file
>> a
=====
H O U S E
=====
(this: 0x6c1a00)
address: Teststrasse 123, 45678 Testort
number of households: 1
total number of consumers: 1
H O U S E H O L D N O 0 P O W E R C O N S U M P T I O N
-----
(this: 0x6c17a0)
```

```

        price for one kWh: 40.00 ct/kWh
        power supplier: RWE
        square metres: 100.00 qm
        persons: 1
    water heated using electricity: yes
        list of devices
-----
                1: E-SUV
                (this: 0x6c5f00)
    power consumption 100 km: 21.50 kW
                km driven: 100.00 km Monday to Friday
        annual consumption: 5590.0 kWh
        annual costs: 2236.00 EUR
                2: Balcony Power Plant
                (this: 0x6c5ce0)
    solar cells installed power: 800.00 watt_peak
        age of solar cells: 10 years
        annual production: -760.0 kWh
        annual savings: -304.00 EUR
-----
    power consumption square meters: 900.0 kWh
    power consumption all persons: 550.0 kWh
    total annual power consumption: 6280.0 kWh
    total annual power costs: 2512.00 EUR
=====
q quit
d delete house
h house initialisation
i input immobile consumer
m input mobile consumer
s input solar producer
u move up device
p print household
a print all households
n new household
c copy all devices (added to already existing ones)
r read data from file
w write data into file
>> w
input file name: house2.csv
input separator character: ;
output file "house2.csv" opened...
output file "house2.csv" closed
q quit
d delete house
h house initialisation
i input immobile consumer
m input mobile consumer
s input solar producer
u move up device
p print household
a print all households
n new household
c copy all devices (added to already existing ones)
r read data from file
w write data into file
>> d
Mobile E-SUV at address 0x6c5f00 deleted
Consumer E-SUV at address 0x6c5f00 deleted
Device E-SUV at address 0x6c5f00 deleted
Solar Balcony Power Plant at address 0x6c5ce0 deleted
Producer Balcony Power Plant at address 0x6c5ce0 deleted
Device Balcony Power Plant at address 0x6c5ce0 deleted
Household at address 0x6c17a0 deleted
House at address 0x6c1a00 deleted
Adresse Teststrasse 123, 45678 Testort at address 0x6c1a00 deleted
q quit
d delete house
h house initialisation
i input immobile consumer
m input mobile consumer
s input solar producer
u move up device
p print household
a print all households
n new household
```



```
c copy all devices (added to already existing ones)
r read data from file
w write data into file
>> a
house is a nullptr, please first choose h to initialise a new house or r to read from file
q quit
d delete house
h house initialisation
i input immobile consumer
m input mobile consumer
s input solar producer
u move up device
p print household
a print all households
n new household
c copy all devices (added to already existing ones)
r read data from file
w write data into file
>>
```

## Beispiel Programmlauf 2/[Example Program Run 2](#)

```
CALCULATION OF AVERAGE POWER COSTS FOR A HOUSE – CLASS VERSION WITH INHERITANCE
q quit
d delete house
h house initialisation
i input immobile consumer
m input mobile consumer
s input solar producer
u move up device
p print household
a print all households
n new household
c copy all devices (added to already existing ones)
r read data from file
w write data into file
>> a
house is a nullptr, please first choose h to initialise a new house or r to read from file
q quit
d delete house
h house initialisation
i input immobile consumer
m input mobile consumer
s input solar producer
u move up device
p print household
a print all households
n new household
c copy all devices (added to already existing ones)
r read data from file
w write data into file
>> r
input file name: house3.csv
input separator character: ;
input file "house3.csv" opened...
Address Lotharstrasse 65d, 47057 Duisburg Neudorf at address 0x62f6c0 deleted
  is an unknown structure
input file "house3.csv" closed
q quit
d delete house
h house initialisation
i input immobile consumer
m input mobile consumer
s input solar producer
u move up device
p print household
a print all households
n new household
c copy all devices (added to already existing ones)
r read data from file
w write data into file
>> a
=====
                        H O U S E
=====
                        (this: 0xf65ef0)
                        address: Lotharstrasse 65d, 47057 Duisburg Neudorf
                        number of households: 6
                        total number of consumers: 7
H O U S E H O L D   N O   1   P O W E R   C O N S U M P T I O N
-----
                        (this: 0xf65fd0)
                        price for one kWh: 30.00 ct/kWh
                        power supplier: ENBW
                        square metres: 100.00 qm
                        persons: 1
                        water heated using electricity: yes
                        list of devices
-----
                        1: E-Bike
                        (this: 0xf66020)
                        power consumption 100 km: 0.50 kW
                        km driven: 100.00 km daily
                        annual consumption: 182.5 kWh
                        annual costs: 54.75 EUR
                        2: balcony power plant
                        (this: 0xf660d0)
```

<div><div>solar cells installed power: 600.00 watt_peak</div><div>age of solar cells: 1 years</div><div>annual production: -597.0 kWh</div><div>annual savings: -179.10 EUR</div></div>
<div><div>power consumption square meters: 900.0 kWh</div><div>power consumption all persons: 550.0 kWh</div><div>total annual power consumption: 1035.5 kWh</div><div>total annual power costs: 310.65 EUR</div></div> <div>H O U S E H O L D   N O   2   P O W E R   C O N S U M P T I O N</div>
<div><div>(this: 0xf66120)</div><div>price for one kWh: 30.00 ct/kWh</div><div>power supplier: Yello Strom</div><div>square metres: 200.00 qm</div><div>persons: 5</div><div>water heated using electricity: yes</div><div>list of devices</div></div>
<div><div>1: balcony power plant</div><div>(this: 0xf66170)</div><div>solar cells installed power: 600.00 watt_peak</div><div>age of solar cells: 1 years</div><div>annual production: -597.0 kWh</div><div>annual savings: -179.10 EUR</div><div>2: Router</div><div>(this: 0xf661f0)</div><div>power consumption: 10.00 W</div><div>power consumption standby: 0.00 W</div><div>annual hours of use: 8760.00 h</div><div>annual hours of standby: 0.00 h</div><div>annual consumption: 87.6 kWh</div><div>annual costs: 26.28 EUR</div><div>3: Office PC</div><div>(this: 0xf66250)</div><div>power consumption: 200.00 W</div><div>power consumption standby: 0.50 W</div><div>annual hours of use: 2210.00 h</div><div>annual hours of standby: 6550.00 h</div><div>annual consumption: 445.3 kWh</div><div>annual costs: 133.58 EUR</div><div>4: Washing Machine</div><div>(this: 0xf662e0)</div><div>power consumption: 2000.00 W</div><div>power consumption standby: 0.00 W</div><div>annual hours of use: 104.00 h</div><div>annual hours of standby: 8656.00 h</div><div>annual consumption: 208.0 kWh</div><div>annual costs: 62.40 EUR</div></div>
<div><div>power consumption square meters: 1800.0 kWh</div><div>power consumption all persons: 2750.0 kWh</div><div>total annual power consumption: 4693.9 kWh</div><div>total annual power costs: 1408.16 EUR</div></div> <div>H O U S E H O L D   N O   3   P O W E R   C O N S U M P T I O N</div>
<div><div>(this: 0xf66340)</div><div>price for one kWh: 40.00 ct/kWh</div><div>power supplier: Stadtwerke</div><div>square metres: 100.00 qm</div><div>persons: 2</div><div>water heated using electricity: no</div><div>list of devices</div></div>
<div><div>1: balcony power plant</div><div>(this: 0xf66390)</div><div>solar cells installed power: 0.80 watt_peak</div><div>age of solar cells: 2 years</div><div>annual production: -0.8 kWh</div><div>annual savings: -0.32 EUR</div><div>2: E-SUV</div><div>(this: 0xf66410)</div><div>power consumption 100 km: 18.00 kW</div><div>km driven: 100.00 km Monday to Friday</div><div>annual consumption: 4680.0 kWh</div><div>annual costs: 1872.00 EUR</div></div>



```

        3: Dish Washer
        (this: 0xf66490)
        power consumption: 250.00 W
    power consumption standby: 0.00 W
        annual hours of use: 1277.50 h
        annual hours of standby: 7482.50 h
        annual consumption: 319.4 kWh
        annual costs: 127.75 EUR
        4: LED TV
        (this: 0xf664f0)
        power consumption: 70.00 W
    power consumption standby: 0.50 W
        annual hours of use: 208.00 h
        annual hours of standby: 8552.00 h
        annual consumption: 18.8 kWh
        annual costs: 7.53 EUR

```

```

power consumption square meters: 900.0 kWh
power consumption all persons: 400.0 kWh
total annual power consumption: 6317.4 kWh
total annual power costs: 2526.97 EUR

```

```

=====
q quit
d delete house
h house initialisation
i input immobile consumer
m input mobile consumer
s input solar producer
u move up device
p print household
a print all households
n new household
c copy all devices (added to already existing ones)
r read data from file
w write data into file
>> n
number of household? 4
how many square metres does the household have? 50
how many persons live in this household? 2
is hot water heated using electricity? (y(es) or n(o)) y
what is the price for one kWh in EUR? 0.5
who is the power supplier? RWE
q quit
d delete house
h house initialisation
i input immobile consumer
m input mobile consumer
s input solar producer
u move up device
p print household
a print all households
n new household
c copy all devices (added to already existing ones)
r read data from file
w write data into file
>> c
number of household from which to copy devices? 2
number of household to copy to? 4
q quit
d delete house
h house initialisation
i input immobile consumer
m input mobile consumer
s input solar producer
u move up device
p print household
a print all households
n new household
c copy all devices (added to already existing ones)
r read data from file
w write data into file
>> c
number of household from which to copy devices? 3
number of household to copy to? 4
q quit
d delete house

```

```
h house initialisation
i input immobile consumer
m input mobile consumer
s input solar producer
u move up device
p print household
a print all households
n new household
c copy all devices (added to already existing ones)
r read data from file
w write data into file
>> a
```

=====

H O U S E

=====

```
(this: 0xf65ef0)
address: Lotharstrasse 65d, 47057 Duisburg Neudorf
number of households: 6
total number of consumers: 13
```

H O U S E H O L D N O 1 P O W E R C O N S U M P T I O N

-----

```
(this: 0xf65fd0)
price for one kWh: 30.00 ct/kWh
power supplier: ENBW
square metres: 100.00 qm
persons: 1
water heated using electricity: yes
list of devices
```

```
1: E-Bike
(this: 0xf66020)
power consumption 100 km: 0.50 kW
km driven: 100.00 km daily
annual consumption: 182.5 kWh
annual costs: 54.75 EUR
2: balcony power plant
(this: 0xf660d0)
solar cells installed power: 600.00 watt_peak
age of solar cells: 1 years
annual production: -597.0 kWh
annual savings: -179.10 EUR
```

```
power consumption square meters: 900.0 kWh
power consumption all persons: 550.0 kWh
total annual power consumption: 1035.5 kWh
total annual power costs: 310.65 EUR
```

H O U S E H O L D N O 2 P O W E R C O N S U M P T I O N

-----

```
(this: 0xf66120)
price for one kWh: 30.00 ct/kWh
power supplier: Yello Strom
square metres: 200.00 qm
persons: 5
water heated using electricity: yes
list of devices
```

```
1: balcony power plant
(this: 0xf66170)
solar cells installed power: 600.00 watt_peak
age of solar cells: 1 years
annual production: -597.0 kWh
annual savings: -179.10 EUR
2: Router
(this: 0xf661f0)
power consumption: 10.00 W
power consumption standby: 0.00 W
annual hours of use: 8760.00 h
annual hours of standby: 0.00 h
annual consumption: 87.6 kWh
annual costs: 26.28 EUR
3: Office PC
(this: 0xf66250)
power consumption: 200.00 W
power consumption standby: 0.50 W
annual hours of use: 2210.00 h
annual hours of standby: 6550.00 h
```

<div><div>annual consumption: 445.3 kWh</div><div>annual costs: 133.58 EUR</div><div>4: Washing Machine</div><div>(this: 0xf662e0)</div><div>power consumption: 2000.00 W</div><div>power consumption standby: 0.00 W</div><div>annual hours of use: 104.00 h</div><div>annual hours of standby: 8656.00 h</div><div>annual consumption: 208.0 kWh</div><div>annual costs: 62.40 EUR</div></div>
<div><div>power consumption square meters: 1800.0 kWh</div><div>power consumption all persons: 2750.0 kWh</div><div>total annual power consumption: 4693.9 kWh</div><div>total annual power costs: 1408.16 EUR</div></div> <div>H O U S E H O L D   N O   3   P O W E R   C O N S U M P T I O N</div>
<div><div>(this: 0xf66340)</div><div>price for one kWh: 40.00 ct/kWh</div><div>power supplier: Stadtwerke</div><div>square metres: 100.00 qm</div><div>persons: 2</div><div>water heated using electricity: no</div><div>list of devices</div></div>
<div><div>1: balcony power plant</div><div>(this: 0xf66390)</div><div>solar cells installed power: 0.80 watt_peak</div><div>age of solar cells: 2 years</div><div>annual production: -0.8 kWh</div><div>annual savings: -0.32 EUR</div><div>2: E-SUV</div><div>(this: 0xf66410)</div><div>power consumption 100 km: 18.00 kW</div><div>km driven: 100.00 km Monday to Friday</div><div>annual consumption: 4680.0 kWh</div><div>annual costs: 1872.00 EUR</div><div>3: Dish Washer</div><div>(this: 0xf66490)</div><div>power consumption: 250.00 W</div><div>power consumption standby: 0.00 W</div><div>annual hours of use: 1277.50 h</div><div>annual hours of standby: 7482.50 h</div><div>annual consumption: 319.4 kWh</div><div>annual costs: 127.75 EUR</div><div>4: LED TV</div><div>(this: 0xf664f0)</div><div>power consumption: 70.00 W</div><div>power consumption standby: 0.50 W</div><div>annual hours of use: 208.00 h</div><div>annual hours of standby: 8552.00 h</div><div>annual consumption: 18.8 kWh</div><div>annual costs: 7.53 EUR</div></div>
<div><div>power consumption square meters: 900.0 kWh</div><div>power consumption all persons: 400.0 kWh</div><div>total annual power consumption: 6317.4 kWh</div><div>total annual power costs: 2526.97 EUR</div></div> <div>H O U S E H O L D   N O   4   P O W E R   C O N S U M P T I O N</div>
<div><div>(this: 0xf61a30)</div><div>price for one kWh: 50.00 ct/kWh</div><div>power supplier: RWE</div><div>square metres: 50.00 qm</div><div>persons: 2</div><div>water heated using electricity: yes</div><div>list of devices</div></div>
<div><div>1: balcony power plant</div><div>(this: 0xf65ea0)</div><div>solar cells installed power: 0.80 watt_peak</div><div>age of solar cells: 2 years</div><div>annual production: -0.8 kWh</div><div>annual savings: -0.40 EUR</div><div>2: E-SUV</div><div>(this: 0xf66910)</div></div>



```

    power consumption 100 km: 18.00 kW
        km driven: 100.00 km Monday to Friday
    annual consumption: 4680.0 kWh
    annual costs: 2340.00 EUR
    3: Dish Washer
    (this: 0xf66960)
    power consumption: 250.00 W
power consumption standby: 0.00 W
    annual hours of use: 1277.50 h
    annual hours of standby: 7482.50 h
    annual consumption: 319.4 kWh
    annual costs: 159.69 EUR
    4: LED TV
    (this: 0xf669c0)
    power consumption: 70.00 W
power consumption standby: 0.50 W
    annual hours of use: 208.00 h
    annual hours of standby: 8552.00 h
    annual consumption: 18.8 kWh
    annual costs: 9.42 EUR
    5: balcony power plant
    (this: 0xf65ce0)
solar cells installed power: 600.00 watt_peak
    age of solar cells: 1 years
    annual production: -597.0 kWh
    annual savings: -298.50 EUR
    6: Router
    (this: 0xf667f0)
    power consumption: 10.00 W
power consumption standby: 0.00 W
    annual hours of use: 8760.00 h
    annual hours of standby: 0.00 h
    annual consumption: 87.6 kWh
    annual costs: 43.80 EUR
    7: Office PC
    (this: 0xf66850)
    power consumption: 200.00 W
power consumption standby: 0.50 W
    annual hours of use: 2210.00 h
    annual hours of standby: 6550.00 h
    annual consumption: 445.3 kWh
    annual costs: 222.64 EUR
    8: Washing Machine
    (this: 0xf668b0)
    power consumption: 2000.00 W
power consumption standby: 0.00 W
    annual hours of use: 104.00 h
    annual hours of standby: 8656.00 h
    annual consumption: 208.0 kWh
    annual costs: 104.00 EUR

```

```

-----
power consumption square meters: 450.0 kWh
power consumption all persons: 1100.0 kWh
total annual power consumption: 6711.3 kWh
total annual power costs: 3355.65 EUR
=====

```

```

q quit
d delete house
h house initialisation
i input immobile consumer
m input mobile consumer
s input solar producer
u move up device
p print household
a print all households
n new household
c copy all devices (added to already existing ones)
r read data from file
w write data into file
>> u
number of household? 4
which one? 8
q quit
d delete house
h house initialisation
i input immobile consumer

```

```
m input mobile consumer
s input solar producer
u move up device
p print household
a print all households
n new household
c copy all devices (added to already existing ones)
r read data from file
w write data into file
>> u
number of household? 4
which one? 7
q quit
d delete house
h house initialisation
i input immobile consumer
m input mobile consumer
s input solar producer
u move up device
p print household
a print all households
n new household
c copy all devices (added to already existing ones)
r read data from file
w write data into file
>> a
```

=====	
H O U S E	
=====	
(this: 0xf65ef0)	
address: Lotharstrasse 65d, 47057 Duisburg Neudorf	
number of households: 6	
total number of consumers: 13	
H O U S E H O L D N O 1 P O W E R C O N S U M P T I O N	
-----	
(this: 0xf65fd0)	
price for one kWh: 30.00 ct/kWh	
power supplier: ENBW	
square metres: 100.00 qm	
persons: 1	
water heated using electricity: yes	
list of devices	
-----	
1: E-Bike	
(this: 0xf66020)	
power consumption 100 km: 0.50 kW	
km driven: 100.00 km daily	
annual consumption: 182.5 kWh	
annual costs: 54.75 EUR	
2: balcony power plant	
(this: 0xf660d0)	
solar cells installed power: 600.00 watt_peak	
age of solar cells: 1 years	
annual production: -597.0 kWh	
annual savings: -179.10 EUR	
-----	
power consumption square meters: 900.0 kWh	
power consumption all persons: 550.0 kWh	
total annual power consumption: 1035.5 kWh	
total annual power costs: 310.65 EUR	
H O U S E H O L D N O 2 P O W E R C O N S U M P T I O N	
-----	
(this: 0xf66120)	
price for one kWh: 30.00 ct/kWh	
power supplier: Yello Strom	
square metres: 200.00 qm	
persons: 5	
water heated using electricity: yes	
list of devices	
-----	
1: balcony power plant	
(this: 0xf66170)	
solar cells installed power: 600.00 watt_peak	
age of solar cells: 1 years	
annual production: -597.0 kWh	
annual savings: -179.10 EUR	

<div>2: Router (this: 0xf661f0) power consumption: 10.00 W power consumption standby: 0.00 W annual hours of use: 8760.00 h annual hours of standby: 0.00 h annual consumption: 87.6 kWh annual costs: 26.28 EUR</div> <div>3: Office PC (this: 0xf66250) power consumption: 200.00 W power consumption standby: 0.50 W annual hours of use: 2210.00 h annual hours of standby: 6550.00 h annual consumption: 445.3 kWh annual costs: 133.58 EUR</div> <div>4: Washing Machine (this: 0xf662e0) power consumption: 2000.00 W power consumption standby: 0.00 W annual hours of use: 104.00 h annual hours of standby: 8656.00 h annual consumption: 208.0 kWh annual costs: 62.40 EUR</div>	
<div>power consumption square meters: 1800.0 kWh power consumption all persons: 2750.0 kWh total annual power consumption: 4693.9 kWh total annual power costs: 1408.16 EUR</div> <div>H O U S E H O L D   N O   3   P O W E R   C O N S U M P T I O N</div>	
<div>(this: 0xf66340) price for one kWh: 40.00 ct/kWh power supplier: Stadtwerke square metres: 100.00 qm persons: 2 water heated using electricity: no list of devices</div>	
<div>1: balcony power plant (this: 0xf66390) solar cells installed power: 0.80 watt_peak age of solar cells: 2 years annual production: -0.8 kWh annual savings: -0.32 EUR</div> <div>2: E-SUV (this: 0xf66410) power consumption 100 km: 18.00 kW km driven: 100.00 km Monday to Friday annual consumption: 4680.0 kWh annual costs: 1872.00 EUR</div> <div>3: Dish Washer (this: 0xf66490) power consumption: 250.00 W power consumption standby: 0.00 W annual hours of use: 1277.50 h annual hours of standby: 7482.50 h annual consumption: 319.4 kWh annual costs: 127.75 EUR</div> <div>4: LED TV (this: 0xf664f0) power consumption: 70.00 W power consumption standby: 0.50 W annual hours of use: 208.00 h annual hours of standby: 8552.00 h annual consumption: 18.8 kWh annual costs: 7.53 EUR</div>	
<div>power consumption square meters: 900.0 kWh power consumption all persons: 400.0 kWh total annual power consumption: 6317.4 kWh total annual power costs: 2526.97 EUR</div> <div>H O U S E H O L D   N O   4   P O W E R   C O N S U M P T I O N</div>	
<div>(this: 0xf61a30) price for one kWh: 50.00 ct/kWh</div>	



```
power supplier: RWE
square metres: 50.00 qm
persons: 2
water heated using electricity: yes
list of devices
-----
1: balcony power plant
  (this: 0xf65ea0)
solar cells installed power: 0.80 watt_peak
age of solar cells: 2 years
annual production: -0.8 kWh
annual savings: -0.40 EUR
2: E-SUV
  (this: 0xf66910)
power consumption 100 km: 18.00 kW
  km driven: 100.00 km Monday to Friday
annual consumption: 4680.0 kWh
annual costs: 2340.00 EUR
3: Dish Washer
  (this: 0xf66960)
power consumption: 250.00 W
power consumption standby: 0.00 W
annual hours of use: 1277.50 h
annual hours of standby: 7482.50 h
annual consumption: 319.4 kWh
annual costs: 159.69 EUR
4: LED TV
  (this: 0xf669c0)
power consumption: 70.00 W
power consumption standby: 0.50 W
annual hours of use: 208.00 h
annual hours of standby: 8552.00 h
annual consumption: 18.8 kWh
annual costs: 9.42 EUR
5: balcony power plant
  (this: 0xf65ce0)
solar cells installed power: 600.00 watt_peak
age of solar cells: 1 years
annual production: -597.0 kWh
annual savings: -298.50 EUR
6: Washing Machine
  (this: 0xf668b0)
power consumption: 2000.00 W
power consumption standby: 0.00 W
annual hours of use: 104.00 h
annual hours of standby: 8656.00 h
annual consumption: 208.0 kWh
annual costs: 104.00 EUR
7: Router
  (this: 0xf667f0)
power consumption: 10.00 W
power consumption standby: 0.00 W
annual hours of use: 8760.00 h
annual hours of standby: 0.00 h
annual consumption: 87.6 kWh
annual costs: 43.80 EUR
8: Office PC
  (this: 0xf66850)
power consumption: 200.00 W
power consumption standby: 0.50 W
annual hours of use: 2210.00 h
annual hours of standby: 6550.00 h
annual consumption: 445.3 kWh
annual costs: 222.64 EUR
-----
power consumption square meters: 450.0 kWh
power consumption all persons: 1100.0 kWh
total annual power consumption: 6711.3 kWh
total annual power costs: 3355.65 EUR
=====
q quit
d delete house
h house initialisation
i input immobile consumer
m input mobile consumer
s input solar producer
```

```
u move up device
p print household
a print all households
n new household
c copy all devices (added to already existing ones)
r read data from file
w write data into file
>> w
input file name: house4.csv
input separator character: |
output file "house4.csv" opened...
output file "house4.csv" closed
q quit
d delete house
h house initialisation
i input immobile consumer
m input mobile consumer
s input solar producer
u move up device
p print household
a print all households
n new household
c copy all devices (added to already existing ones)
r read data from file
w write data into file
>>
```

Zuletzt geändert: Donnerstag, 18. Januar 2024, 00:47

◀ A5 Upload Teil 1/Part 1

Direkt zu:

house1.csv ►

- Deutsch (de)
- Dansk (da)
- Deutsch (de)
- English (en)
- Español - España (es\_es)
- Español - Internacional (es)
- Français (fr)
- Polski (pl)
- Türkçe (tr)
- Русский (ru)
- Українська (uk)

Moodle an der UDE ist ein Service des ZIM  
Datenschutzerklärung | Impressum | Kontakt