**I4 Principy objektového programování. Vlastnosti jazyka C++, datové proudy, STL kontejnery, šablony, zpracování výjimek. (Programování v C)**

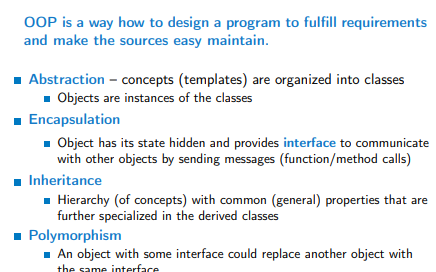
**Principy objektového programování**

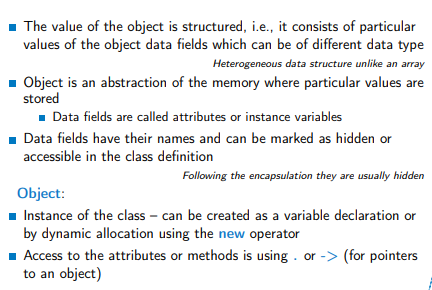
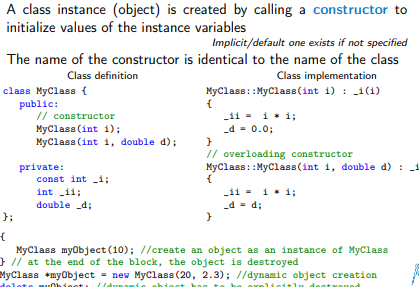
Výkonný kód je v objektovém programování přidružen k datům (metody jsou zapouzdřeny v objektech), což umožňuje snadnější přenos kódu mezi různými projekty (abstrakce a zapouzdření).

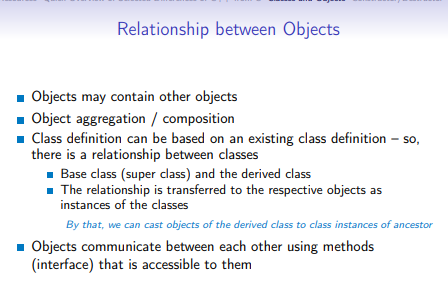
**Objekty** – jednotlivé prvky modelované reality (jak data, tak související funkčnost) jsou v programu seskupeny do entit, nazývaných objekty. Objekty si pamatují svůj stav a navenek poskytují operace (přístupné jako metody pro volání).

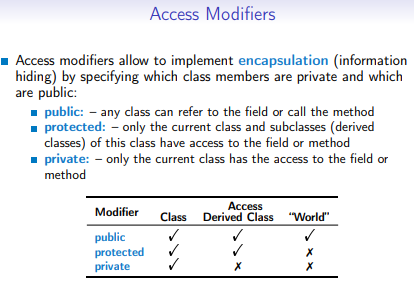
* má atributy (stav) a metody (chování)

**Třída (class)** – Třída je schéma, které popisuje vnitřní strukturu objektu a jeho vnější rozhraní. Na základě třídy je možné tvořit jednotlivé objekty.

OOP is a way how to design a program to fulfill requirements and make the sources easy maintain. 



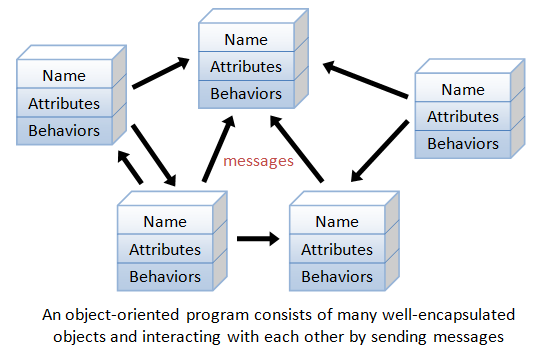




Funkce objektů spočívá v tom, že vlastní procedury objektu mohou přistupovat k datovým polím sebe sama a často je upravovat (objekty mají představu o this nebo o self). V OOP jsou počítačové programy navrženy tak, že jsou vyrobeny z objektů, které na sebe vzájemně působí. Jazyky OOP jsou různorodé, ale nejoblíbenější jsou založené na třídách, což znamená, že objekty jsou instancemi tříd, které také určují jejich typy.

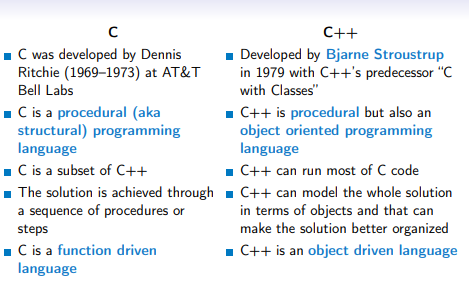
Objekty – jednotlivé prvky modelované reality (jak data, tak související funkčnost) jsou v programu seskupeny do entit, nazývaných objekty. Objekty si pamatují svůj stav a navenek poskytují operace (přístupné jako metody pro volání).

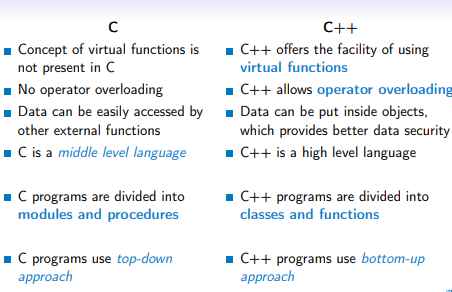
Jazyky - Java, C++, C#, Python, R, PHP, Visual Basic.NET, JavaScript, Ruby, Perl, Object Pascal, Objective-C, Dart, Swift, Scala, Kotlin, Common Lisp, MATLAB, and Smalltalk.

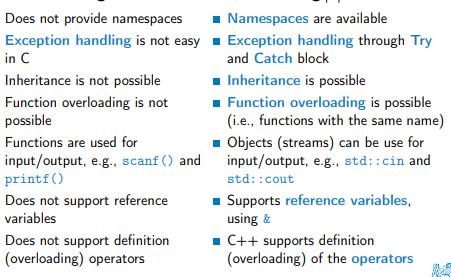


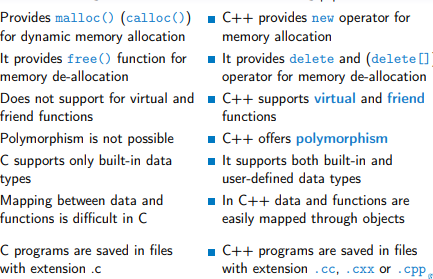
## Vlastnosti jazyka C++

* V současné době patří C++ mezi nejrozšířenější programovací jazyky.
* Primitivní typy: char, int, short, long, unsigned \_\_, float, double, apod.
* jazyk vyšší úrovně
* zdrojový kód mívá příponu .cpp
* hlavičkové soubory mívají příponu .h









## Datové proudy

Datové proudy (anglicky stream; dále jen „proudy“) jsou sekvence dat. Proud je definován svým vstupem a výstupem, těmi mohou být například soubor na disku, zařízení (vstupní klávesnice nebo myš, výstupní displej) nebo jiný program. Primitivní proudy pouze přenášejí binární data, specializované proudy umí s přenášenými daty manipulovat. Tyto manipulační schopnosti se liší podle typu přenášených dat.

In connection-oriented communication, a data stream is a sequence of digitally encoded coherent signals (packets of data or data packets) used to transmit or receive information that is in the process of being transmitted.[1] A data stream is a set of extracted information from a data provider.[2] It contains raw data that was gathered out of users' browser behavior from websites, where a dedicated pixel is placed. Data streams are useful for data scientists for big data and AI algorithms supply

## STL kontejnery

Standard Template Library (STL) is a library of the standard C++ that provides efficient implementations of the data containers, algorithms, functions, and iterators

High efficiency of the implementation is achieved by templates with compile-type polymorphism ([*Polymorfismus*](https://cs.wikipedia.org/wiki/Polymorfismus_(programov%C3%A1n%C3%AD)) – odkazovaný objekt se chová podle toho, jaké třídy je instancí)

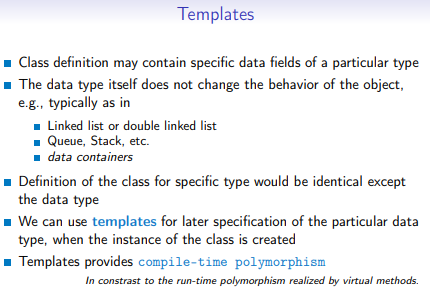
Jedná se o standardní knihovnu šablon jazyka C++. Knihovna by měla být dodávána s každým překladačem C++. V STL můžeme najít mnoho různých a užitečných šablon. Mimo jiné zde jsou šablony datových kontejnerů, iterátory a šablony algoritmů, které pomocí iterátorů pracují s datovými kontejner.

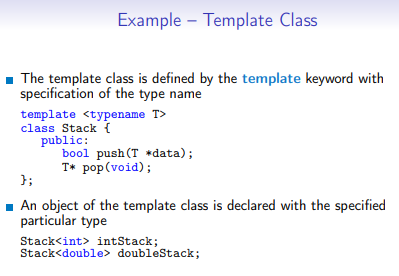
Příklady těchto datových kontejnerů: *list*, *map*, *queue* (fronta), *stack* (zásobník). Jsou to vlastně šablony tříd.

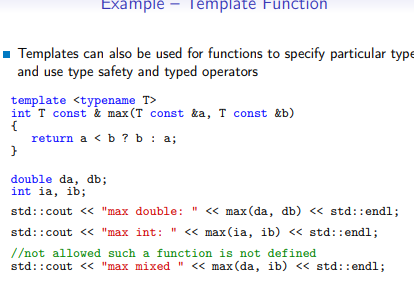
Pod pojmem datový kontejner si představme něco, co má schopnost "zapamatovat si" nějaké data. Data jsou v kontejneru nějakým způsobem uloženy. kontejnery v STL se dělí na posloupnosti a asociativní kontejnery. Posloupnosti jsou logické posloupnosti dat, které ale nemusejí být v paměti fyzicky za sebou. Asociativní kontejnery jsou kontejnery, kde k datům přistupujeme pomocí nějakého klíče

## Šablony

Jazyk C++ poskytuje možnost napsat tzv. šablonu funkce (template function, generickou funkci) – tj. kód bez specifikace konkrétního typu parametrů a výsledku (nespecifikovaný parametr je generický parametr). Konkrétní funkce (generická instance) se vytvoří automaticky při překladu (při volání se skutečnými parametry).

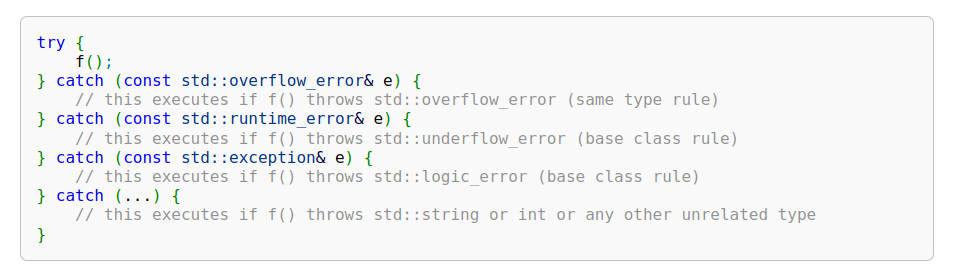






## Výjimky

Výjimka znamená, že se stalo něco chybového, co prostě nepatří do správného chodu programu. Například, pokud ti praskne guma u auta, není extra moudré pokračovat v jízdě. Musíš buď nasadit novou gumu, nebo úplně ukončit styk s vozovkou a nechat si ho odtáhnout.

V C++ a podobných jazycích jsou vyjímky reprezentovány třídami a můžou znamenat mnohé chyby - třeba overflow i underflow proměnné, nějaké logické výjimky, chyby u datových typů, paměti a mnoho dalšího. Navíc si můžeme svoje vlastní výjimky jednoduše přidat vytvořením nové třídy. Když pak máme ve funkci nějakou podmínku a ta není splněna, použijeme **throw JménoVýjimky**. Takový to mechanismus je normálně používaný v celé standardní knihovně. No a když máme kód, u kterého očekáváme možné výjimky, použijeme **try-catch** blok. Tento kód dáme do **try** bloku. U **catch** bloku pak dáme vždy jméno výjimky do závorky a uvnitř kód pro zpracování konkrétní výjimky. Pro každou výjimku je pak jeden catch blok. V catch bloku může být teda kód co udělá něco abychom se problému zbavili a nebo prostě ukončí program a čus, to už je na nás.

Tento obrázek hezky ilustruje. Pokud se ve funkci f někde vykoná **throw JménoVýjimky**), pak se vykoná kód v příslušném catch bloku. Pokud je throw zavoláno mimo try blok, program jednoduše končí.

Další příklad s názornějším kódem.

**

## Vlastní výjimky je dobré odvodit ze standardní třídy std::exception.