

Cvičení 03 – dědičnost

Jednotlivé třídy realizujte jako samostatné CPP a H soubory!

- **Třída *Objekt***
 - Atributy: *int id*, *double x*, *double y*
 - Parametrický konstruktor: *id*
 - Neparametrický konstruktor nevytvářet!
 - Definujte virtuální destruktork: *virtual ~Objekt() { }*
 - Getter/setter: *x*, *y*
 - Getter: *id*
- **Třída *StatickyObjekt***
 - Potomek *Objekt*
 - Atribut: *TypPrekazky typPrekazky* (výčtový typ – Skala, MalaRostlina, VelkaRostlina)
 - Parametrický konstruktor (*id*, *typPrekazky*)
 - Getter: *typPrekazky*
- **Třída *PohyblivyObjekt***
 - Potomek *Objekt*
 - Atribut: *double uhelNatoceni* (getter, setter; hodnota 0 – 2* π , 0 – sever, π - jih)
- **Třída *Monstrum***
 - Potomek *PohyblivyObjekt*
 - Atribut: *int hp* (getter, setter), *int maxhp* (getter, setter)
- **Třída *Hra***
 - Atribut: *Objekt** objekty*
 - Dynamicky alokované pole ukazatelů na *Objekt*
 - Metody:
 - *void vložObjekt(Objekt* o)*
Vloží objekt do pole.
 - *int* najdiIdStatickychObjektu(double xmin, double xmax, double ymin, double ymax)*
Vrátí pole id všech statických objektů v zadaných souřadnicích.
 - *PohyblivyObjekt** najdiPohybliveObjektyVOblasti(double x, double y, double r)*
Vrátí pole ukazelů na pohyblivé objekty v kruhové oblasti (S=[x,y], radius=r).
 - *PohyblivyObjekt** najdiPohybliveObjektyVOblasti(double x, double y, double r, double umin, double umax)*
Viz předchozí, navíc kontrola úhlu natočení.

Jak otestovat, jestli je v *Objekt* o* potomek?

```
Objekt* objekt = new StatickyObjekt{1, TypPrekazky::MalaRostlina};
StatickyObjekt* so = dynamic_cast<StatickyObjekt*>(objekt);
if (so != nullptr)
    cout << „objekt je StatickyObjekt (nebo jeho potomek)“ << so->getTypPrekazky();
```

Vypracovaný projekt odevzdejte do konce cvičení na STAG! Ve formátu ZIP.