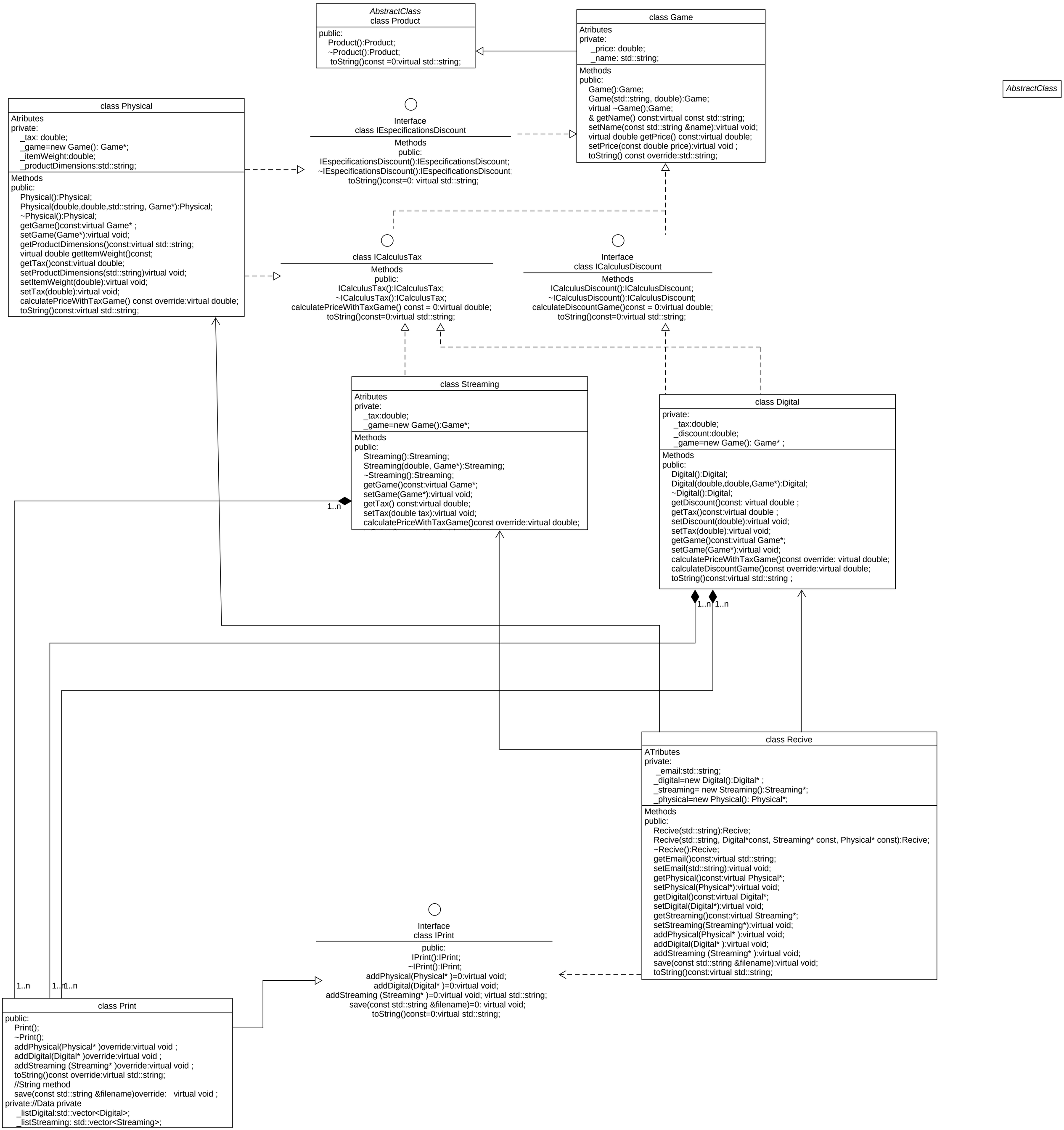
AbstractClass
class Product
public:
Product():Product;
~Product():Product;
toString()const =0:virtual std::string;
class Game
Atributes
private:
_price: double;
_name: std::string;
Methods
public:
Game():Game;
Game(std::string, double):Game;
virtual ~Game();Game;
& getName() const:virtual const std::string;
setName(const std::string &name):virtual void;
virtual double getPrice() const:virtual double;
setPrice(const double price):virtual void ;
toString() const override:std::string;
AbstractClass
class Physical
Atributes
private:
_tax: double;
_game=new Game(): Game*;
_itemWeight:double;
_productDimensions:std::string;
Methods
public:
Physical():Physical;
Physical(double,double,std::string, Game*):Physical;
~Physical():Physical;
getGame()const:virtual Game* ;
setGame(Game*):virtual void;
getProductDimensions()const:virtual std::string;
virtual double getItemWeight()const;
getTax()const:virtual double;
setProductDimensions(std::string)virtual void;
setItemWeight(double):virtual void;
setTax(double):virtual void;
calculatePriceWithTaxGame() const override:virtual double;
toString()const:virtual std::string;
Interface
class IEspecificationsDiscount
Methods
public:
IEspecificationsDiscount():IEspecificationsDiscount;
~IEspecificationsDiscount():IEspecificationsDiscount;
toString()const=0: virtual std::string;
class ICalculusTax
Methods
public:
ICalculusTax():ICalculusTax;
~ICalculusTax():ICalculusTax;
calculatePriceWithTaxGame() const = 0:virtual double;
toString()const=0:virtual std::string;
Interface
class ICalculusDiscount
Methods
ICalculusDiscount():ICalculusDiscount;
~ICalculusDiscount():ICalculusDiscount;
calculateDiscountGame()const = 0:virtual double;
toString()const=0:virtual std::string;
class Streaming
Atributes
private:
_tax:double;
_game=new Game():Game*;
Methods
public:
Streaming():Streaming;
Streaming(double, Game*):Streaming;
~Streaming():Streaming;
getGame()const:virtual Game*;
setGame(Game*):virtual void;
getTax() const:virtual double;
setTax(double tax):virtual void;
calculatePriceWithTaxGame()const override:virtual double;
1..n
class Digital
private:
_tax:double;
_discount:double;
_game=new Game(): Game* ;
Methods
public:
Digital():Digital;
Digital(double,double,Game*):Digital;
~Digital():Digital;
getDiscount()const: virtual double ;
getTax()const:virtual double ;
setDiscount(double):virtual void;
setTax(double):virtual void;
getGame()const:virtual Game*;
setGame(Game*):virtual void;
calculatePriceWithTaxGame()const override: virtual double;
calculateDiscountGame()const override:virtual double;
toString()const:virtual std::string ;
1..n
1..n
class Recive
ATributes
private:
_email:std::string;
_digital=new Digital():Digital* ;
_streaming= new Streaming():Streaming*;
_physical=new Physical(): Physical*;
Methods
public:
Recive(std::string):Recive;
Recive(std::string, Digital*const, Streaming* const, Physical* const):Recive;
~Recive():Recive;
getEmail()const:virtual std::string;
setEmail(std::string):virtual void;
getPhysical()const:virtual Physical*;
setPhysical(Physical*):virtual void;
getDigital()const:virtual Digital*;
setDigital(Digital*):virtual void;
getStreaming()const:virtual Streaming*;
setStreaming(Streaming*):virtual void;
addPhysical(Physical*):virtual void;
addDigital(Digital*):virtual void;
addStreaming (Streaming*):virtual void;
save(const std::string &filename):virtual void;
toString()const:virtual std::string;
Interface
class IPrint
public:
IPrint():IPrint;
~IPrint():IPrint;
addPhysical(Physical*)=0:virtual void;
addDigital(Digital*)=0:virtual void;
addStreaming (Streaming*)=0:virtual void; virtual std::string;
save(const std::string &filename)=0: virtual void;
toString()const=0:virtual std::string;
1..n
1..n
1..n
class Print
public:
Print();
~Print();
addPhysical(Physical*)override:virtual void ;
addDigital(Digital*)override:virtual void ;
addStreaming (Streaming*)override:virtual void ;
toString()const override:virtual std::string;
//String method
save(const std::string &filename)override: virtual void ;
private://Data private
_listDigital:std::vector<Digital>;
_listStreaming: std::vector<Streaming>;